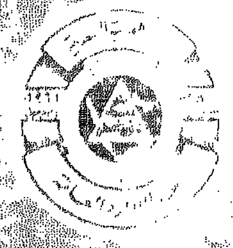
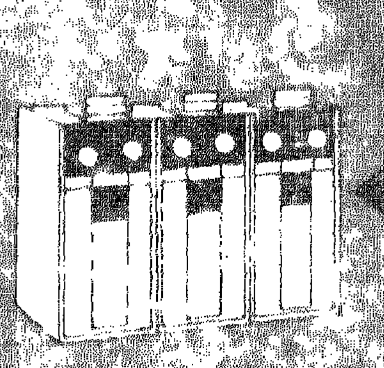
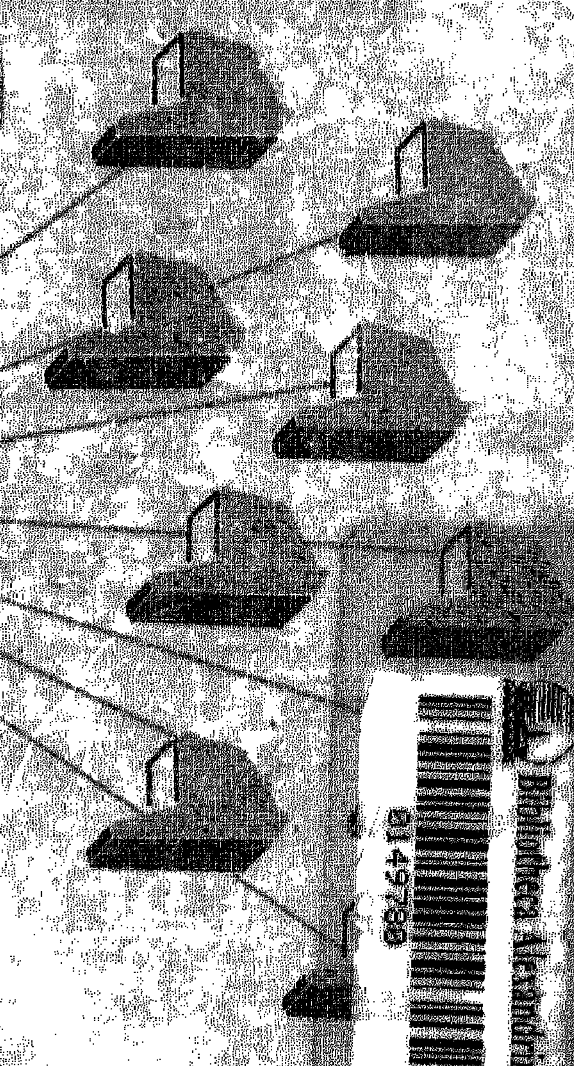


أساليب البرمجة الحديثة بلغة بيسك





تأليف :

الدكتور غازي إسحق الخطيب

عضو هيئة التدريس بالمعهد سابقاً

عزمي حسن العيسى

عضو هيئة التدريس بالمعهد



الإدارة العامة للبحوث

أساليب البرمجة الحديثة بلغة بيسك

تأليف:

صبري محسن العيسى

عضو هيئة التدريس بالمعهد

١٤٠٨هـ/١٩٨٨م

المحتويات

الصفحة

الصفحة

الفصل الثانى عشر: المصفوفات ذات البعدين (Two Dimensional Arrays) ومكوناتها وتعليمات الدوارة المتقدمة

الصفحة

الجزء الثالث: مواضيع متقدمة فى لغة بيسك ٣٧٩

الفصل الثالث عشر: استخدام الملفات فى معالجة البيانات

File Processing ٣٨١

قائمة الأشكال

تابع قائمة الأشكال

قائمة الجداول

مقدمة الكتاب

زاد الحديث فى السنوات الأخيرة عن أهمية الأساليب الحديثة فى البرمجة، وبشكل خاص البرمجة التركيبية والهيكل الهرمى؛ لما لذلك من أثر كبير فى زيادة إنتاجية المبرمج وتحسين وضوح وصيانة البرامج . كذلك فإن لغة بيسك أصبحت شائعة الاستعمال، بسبب كونها اللغة الأساسية التى تتعامل معها الحاسبات الشخصية واسعة الانتشار. لذلك يهدف هذا الكتاب إلى تقديم لغة بيسك وطرق استخدامها، وفق الأساليب الحديثة للبرمجة، ولتحقيق هذا الهدف، تم استخدام برنامج BASICA على جهاز IBM/XT، حيث تتم معالجة البيانات المعطاة باللغة العربية بواسطة تعليمات لغة بيسك باللغة الإنجليزية .

يقع الكتاب فى ستة عشر فصلا، جزئت إلى أجزاء رئيسية ثلاثة، بالإضافة إلى ملحق عن عمليات تشغيل لغة بيسك على بعض الحاسبات الشخصية الكبيرة، بحيث يطلع المستخدم على الأجزاء التى تهمه، وتناسب مستواه فقط . فلم يتم مزج شروحات الأساليب الحديثة وتعليمات البرمجة، بينما تم التقديم لكيفية عمل بعض التعليمات قبل شرحها واستخدامها .

ويمكن تقسيم الكتاب إلى الأجزاء التفصيلية التالية :

١) مقدمة عن الحاسب الآلى : وظائفه وأجزائه.

٢) مقدمة عن الأساليب الحديثة.

٣) مقدمة عن كيفية معالجة البيانات من قبل الحاسب فى فصل مستقل، ومقدمات أخرى متناثرة قبل كل فصل أو موضوع له علاقة بالبرمجة، يستطيع ذو الخبرة فى

البرمجة أن يتجاهلها و يستمر فى القراءة دون أن يعطل ذلك الفهم والاستيعاب لتعليمات اللغة.

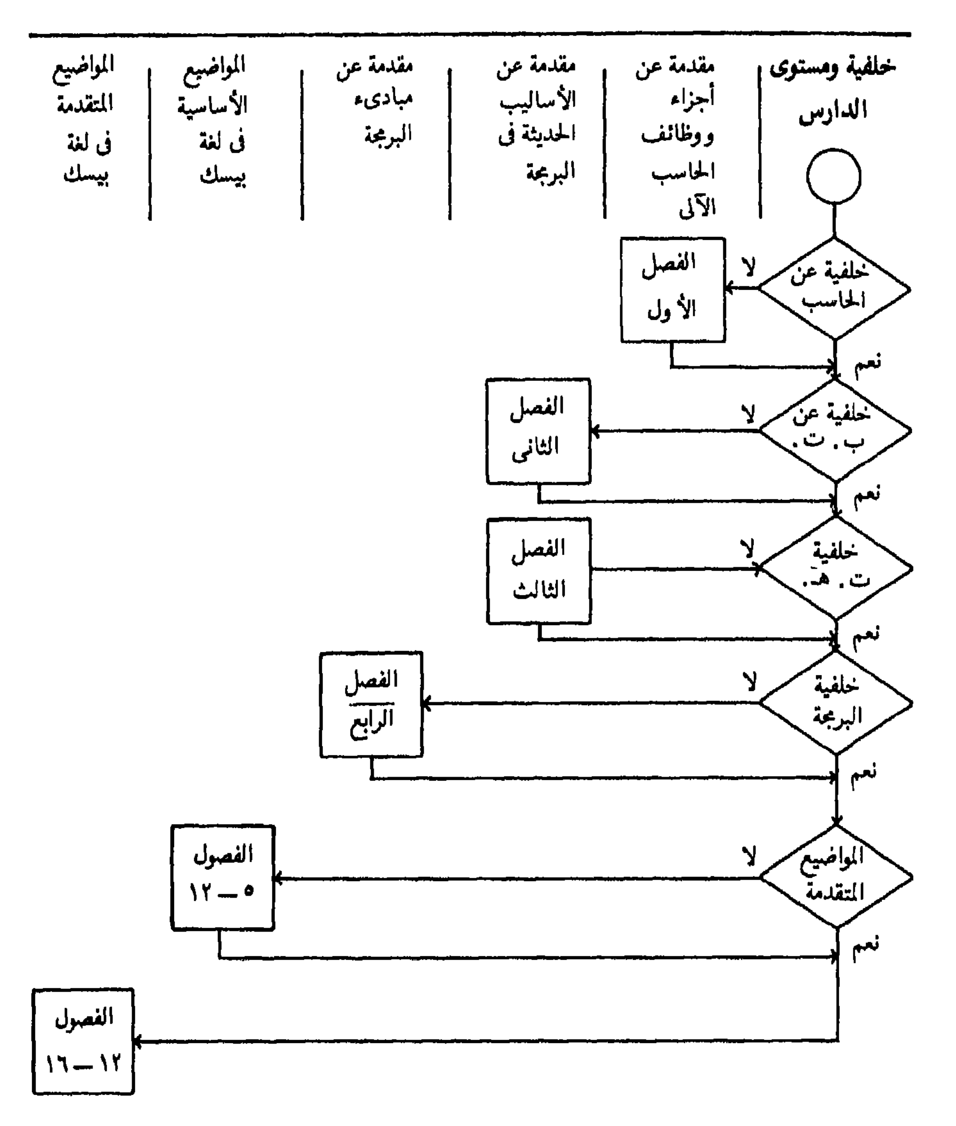
٤) مواضيع أساسية فى كيفية استخدام تعليمات لغة بيسك.
٥) مواضيع متقدمة فى كيفية استخدام تعليمات لغة بيسك.
٦) ملحق تعليمات التشغيل لبعض الحاسبات الكبيرة والصغيرة.

و يبين الرسم التالى كيفية اختيار مواضيع الكتاب للقراءة حسب مستوى وخلفية الدارس، فيما عدا الملحق، حيث يرجع إليه للحصول على تعليمات تشغيل لغة BASICA على جهاز IBM/PC، ولمعرفة الفروقات بين عدة أجهزة أخرى، إلى جانب IBM، فى كيفية استخدامها للغة بيسك (BASIC)، ولمقارنة لغة بيسك بكوبول وفورتران .

ويختلف أسلوب تقديم تعليمات لغة بيسك فى هذا الكتاب عن غيره من الكتب الأخرى* . والأسلوب التقليدى هو التحدث عن قواعد التعليمات وأشكالها المختلفة . والصعوبة التى تواجه الدارس هى فى كيفية ربط خطوات حل مشكلة بتسلسل التعليمات الخاصة بحلها، وكثيراً ما نسمع من الدارسين من يقول : «أنا أفهم كيف تعمل تعليمة اطبع PRINT أو أدخل INPUT إلا أننى أشعر بالجهل التام عندما أحاول ربط هذه التعليمات بأسلوب وخطوات حل مشكلة معينة» .

أما هذا الكتاب فيبدأ من أمثلة بسيطة ومن ثم يقدم التعليمات الضرورية لحل المثال، وأحياناً قد يقدم لضرورة وجود تعليمة ذات وظيفة معينة من واقع أمثلة عملية . وبعد ذلك يتم شرح الأشكال الأخرى للتعليمة ومع أمثلة أخرى أكثر تعقيداً، إذا اقتضى الأمر.

* لقد جاء هذا الكتاب كثمرة لما يزيد على ١٥ سنة من الخبرة التدريبية والتعليمية والعملية للمؤلفين، فى مجال تدريس لغة بيسك وكذا فى مجالى التحليل والتصميم وقواعد البيانات أيضاً .



ب . ت. : برمجة تركيبية

ت . هـ. : تجزئة هرمية وظيفية (هيكل هرمى)

لذا يمتاز أسلوب هذا الكتاب عن غيره من الكتب الأخرى بالخصائص التالية :

١ ــ وجود فصول متقدمة ومستقلة لشرح أسس استخدام الحاسب والأساليب الحديثة فى البرمجة ، بحيث يمكن استخدام هذه الفصول بشكل مستقل عند التدرب على أية لغة برمجة .

٢ ــ الانطلاق من التمارين والأمثلة وخطوات الحل إلى تعليمات بيسك فى توضيح كيفية استخدام الأساليب الحديثة فى البرمجة . وذلك حتى يتمكن الدارس من استخدام هذه التعليمات فى تطبيقات مختلفة .

٣ ــ شرح تعليمة الذهاب إلى برنامج فرعى GOSUB فى الفصول الأولى ، وذلك حتى يعتاد الدارس على كيفية استخدامها فى برمجة أجزاء الهيكل الهرمى .

٤ ــ إجراء التعديلات والتطويرات (الصيانة) على البرامج المستخدمة لحل الأمثلة ، وذلك بتغيير أو تعقيد إجراءات وخطوات الحل فيها ؛ وذلك لإعطاء الدارس فكرة عن كيفية وأهمية إجراء هذه التعديلات على البرامج القائمة .

وقد اتبع الكتاب الخطوات الرئيسية التالية فى معظم الحالات ، إلا إذا اقتضت الضرورة المزج بين هذه الخطوات ، إما لتشابه الأمثلة أو لزيادة توضيحها :

أولاً ــ بعد تحديد الهدف ، يتم سرد الخطوات الرئيسية اللازمة لتحقيق الهدف ، وأحياناً قد يربط بها تعليمات البرمجة المناسبة .

ثانياً ــ الهيكل الهرمى للبرنامج (TOP-DOWN) .

ثالثاً ــ نمط البرمجة التركيبية المناسب .

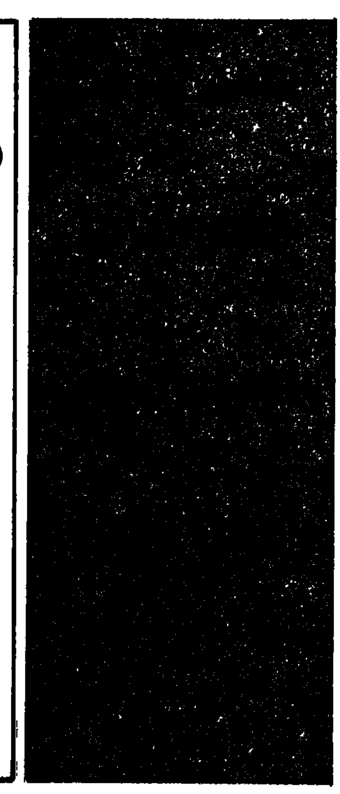
رابعاً ــ البرنامج الكامل وفق أساليب البرمجة الحديثة .

ولا يتطلب استخدام هذا الكتاب بالشكل الأمثل أية خلفيات أو معلومات مسبقة .

ونـود أخـيـراً أن نوجه الشكر إلى زوجتينا وأولادنا الذين ظلوا حبيسى المسكن نهاراً وليـلاً فى ظـروف كـانـوا هـم فـيـهـا أحـوج مـا يكونون إلى أن يمضوا بعض ذلك الوقت خارجه ، وذلك كله حتى نستطيع إنجاز هذا الكتاب فى المدة المحددة .



مقدمة عامة
الحاسب والبرمجة



الحاسب الآلى
(العناصر والوظائف) .
مقدمة عن البرمجة
التركيبية الحديثة.
مقدمة عن التجزئة
الوظيفية الهيكلية(الهرمية)
مقدمة عن وسائل تصنيف
البيانات والمعلومات
فى الحاسب.

الفصل الأول

الحاسب الآلى
العناصر والوظائف

وظائف الحاسب الآلى :

أراد الإنسان منذ القدم أن يستعمل أجهزة تساعده فى القيام ببعض الأعمال التى يقوم بها . وقد أولى الإنسان اهتماماً محدوداً فى العصور القديمة لعمليات العد الحسابى، فكان استخدامه للأصابع والأحجار مثلاً ، ولكن الاهتمام الأكبر انصب على الأعمال التى لها علاقة بالأمور المعيشية والحياتية ، كتوفير المأكل والمشرب والمسكن . ونتج عن ذلك الثورتان الزراعية والصناعية اللتان أدتا إلى استعمال الآلات لتحسين وزيادة الإنتاج الزراعى والصناعى .

ومع التقدم العلمى فى الزراعة والصناعة ، زاد التقدم العمرانى واتصال الأمم بعضها مع بعض، وتعددت النشاطات والعلاقات بين الأفراد والمؤسسات والأمم ، وزادت المهام والوظائف التى تقوم بها الدول والخدمات التى تقدمها لرعاياها ، فكان لابد من الحفاظ على معلومات عن هذه النشاطات والوظائف والخدمات ، بهدف تقويم نتائجها والتخطيط لمستقبلها .

وأول استخدام رئيسى للحاسب الآلى كان فى تعداد السكان للولايات المتحدة الأمريكية فى أواخر القرن التاسع عشر . وبالمقارنة بهذا المثال الحكومى، فإن أول استخدام تجارى كان فى مجال المحاسبة ، حيث تطلب الأمر الحفاظ على معلومات عن العمليات الحاسبية بين صرف وإيراد واستهلاك ، وذلك حتى تعكس الصورة السليمة للوضع المالى للمؤسسات .

و يلاحظ من هذين الاستعمالين الأمور التالية :

١ ــ كثرة المعلومات المراد التعامل معها .

٢ ــ الحاجة إلى القيام بالعمليات الحسابية المطولة .

٣ ــ ضرورة القيام بالمقارنات بهدف التصنيف والفهرسة والتجميع والفرز .

٤ ــ الدقة والسرعة فى إنجاز ماتقدم .

و بناء على ذلك فإن الحاسب يقوم بوظيفتين رئيسيتين بسرعة ودقة متناهيتين :

١ ــ العمليات الحسابية Arithmetic operations : يقوم الحاسب بجميع العمليات الحسابية من جمع وطرح وضرب وقسمة وأس ، مع العلم بأن الحاسب يقوم بجميع العمليات الحسابية بأسلوب الجمع .

٢ ــ العمليات المنطقية Logical operations : يقوم الحاسب بالعمليات المنطقية عن طريق إجراء المقارنات الثنائية بين معلومتين محددتين ، بحيث لاتخرج نتيجة المقارنة فى نطاق محدد عن أحد اختيارين . فمثلا عند مقارنة تساوى معلومتين ، فنتيجة المقارنة تكون إما تساوى كلتيهما أو عدمه ، ولا يندرج صغر أو كبر إحداهما عن الأخرى تحت نطاق هذه المقارنة . وكذلك الأمر مع مقارنات «أصغر من» ، «أكبر من» «لا يساوى» و «أصغر من و يساوى» و «أكبر من و يساوى» . وتستخدم المقارنات هذه فى كثير من الأحيان للتشعب من خطوة إلى أخرى .

ولا بد من التنو يه هنا بأن الحاسب يقوم بواحدة من العمليات الحسابية والمنطقية فقط فى أية لحظة من اللحظات الزمنية التى تمر بها العمليات المتعددة . إذ قد يتطلب استخراج نتائج مطلب معين ، القيام بعمليات حسابية مطولة ومقارنات متعددة ، فالحصول على أسماء الأفراد الذين يتجاوزون الستين من العمروعددهم ،يتطلب التأكد من العمر عن طريق المقارنة المنطقية ، ومن ثم حساب العدد الكلى بعد كل مقارنة إلى

أن تنتهى قائمة أفراد السكان . وقد يصل التعقيد والتداخل بين العمليات الحسابية والمنطقية درجة يصعب معها ملاحظة تسلسل العمليات يدو ياً .

واستناداً إلى الوظيفتين الرئيسيتين : الحسابية والمنطقية ، قد يقوم الحاسب بالوظائف الفرعية التالية :

١ ـ **التجميع والاختصار** : إذ يمكن مثلاً الحصول على مجموع المبالغ المصروفة للرواتب ، ومجموع المبيعات لفترة سنة لمؤسسة ما ، أو مجموع عدد الموظفين فى الدرجات المختلفة . وقد يكون التجميع والاختصار قد بنيا على مقارنات منطقية ، فمن الممكن مثلاً الحصول على عدد الأفراد الذين يقعون ضمن فئات محددة من الأعمار .

٢ ـ **التبويب والتصنيف** : إذ يمكن الحصول على المعلومات حسب أنواع البضائع المنتجة ، أو حسب المواد والقطع المخزنة ، أو حسب أنواع الحسابات ، أو حسب فئات المستهلكين ، أو حسب المواد الداخلة فى التصنيع لسلعة ما .

٣ ـ **الفرز والدمج** : إذ يمكن الحصول على قائمة مسلسلة حسب الحروف الأ بجدية ، أو حسب القيم الرقمية تصاعدياً أو تنازلياً . وكذلك من الممكن دمج قائمتين مفروزتين إذا تشابهت طبيعة المعلومات حرفياً أو رقمياً ، وذلك بهدف إنتاج قائمة جديدة موحدة .

وقد تكون القائمة المفروزة بها أسماء موظفين ، أو الدرجات ومسمياتها ، أو المبيعات حسب المناطق ، أو أرقام القطع وكمياتها المخزنة ، أو درجات الدارسين فى امتحان ما .

٤ ـ **الفهرسة والتقسيم** : إذ يمكن الحصول على المعلومات الخاصة بالسكان مقسمة حسب المناطق الجغرافية ، أو مجمل دخل مندو بى المبيعات حسب نوع السلعة التى يتعاملون معها ، أو تكلفة بناء المنازل حسب تصاميمها المختلفة .

و يلاحظ الفرق بين الفهرسة والتقسيم من جهة، والتبو يب والتصنيف من جهة أخرى : إن الأ ولى لا تتطلب وجود علاقة منطقية تر بط بين الأقسام المختلفة، بينما تتطلب الثانية علاقة يحكمها نظام معين للربط بين الأقسام المختلفة .

فالتصنيف قد يكون بناءً على علاقة هرمية، فتصنيف المكتبات يقسم موضوعات الكتب إلى عشرة أقسام رئيسية، ومن ثم يتم تصنيف الموضوعات الفرعية ضمن كل قسم رئيسى . فإذا أخذنا العلوم الاجتماعية كأحد الأقسام الرئيسية، فتندرج تحته، مثلا : العلوم الاقتصادية، وعلم النفس، وإدارة الأعمال، والإدارة العامة . و يشير التصنيف إلى مستو يات تفصيلية حتى آخر مستوى .

أما تقسيم المساحات إلى مناطق جغرافية لا رابط بينها، فهو فهرسة وليس تصنيفاً، أو قد ترتب المناطق بطريقة عشوائية، أو تفرز حسب أرقامها أو أسمائها .

وحتى يؤدى الحاسب الهدف من استخدامه، لابد من قيامه بوظائف أخرى تكمل الوظائف المذكورة سابقاً .

وهذه الوظائف هى :

١ ـ **عمليات الإدخال** : إذ يجب أن يتم إدخال المعلومات والتعليمات الخاصة بهذه المعلومات، والتى تحدد الخطوات التى ستخضع لها من عمليات حسابية ومنطقية، وغيرها .

٢ ـ **عمليات التخزين** : وذلك لتخزين التعليمات والمعلومات المتعلقة بوصف نشاط معين . و يتم التخزين داخل الحاسب فى نوعين من الذاكرة :

أ ـ الذاكرة الرئيسية (**Main Memory**) : وهى الذاكرة النشطة والمستعدة باستمرار لاستقبال المعلومات والتعليمات طالما كان الحاسب موصولاً بمصدر لتيار

كهربائى . وحال تخزين هذه المعلومات والتعليمات ، تكون قابلة لاتخاذ قرار فورى بتشغيلها من قبل الحاسب .

ب ـ **الذاكرة الفرعية (المساعدة) (Secondary Memory)** : نجمت الحاجة إلى هذا النوع من الذاكرة عن تضخم حجم الذاكرة الرئيسية مع ازدياد استخدام الحاسب فى مجالات كثيرة ، مما أدى إلى ازدياد التكلفة وصعوبة التحكم فى التعليمات والمعلومات المختلفة ، و بالإضافة إلى ذلك فإن بعض التعليمات والمعلومات لانحتاج إليها إلا على فترات متباعدة ، لذلك كان لابد من إيجاد ذاكرة مساعدة يتم فيها تخزين التعليمات والمعلومات إلى حين الحاجة إليها ، فحينئذ تنقل إلى الذاكرة الرئيسية .

٣ ـ **عمليات الإخراج** : وهى العمليات التى لها علاقة بإخراج نتائج تنفيذ العمليات بطريقة تؤدى إلى الاستفادة من وظائف الحاسب من قبل المستخدمين .

ولنضرب مثلاً بسيطاً على هذه الوظائف . لنفترض أن شخصاً ما يريد أن يحسب مساحة السجاد المعد للبيع . لتحقيق ذلك لابد من اتباع الخطوات التالية :

١ ـ الحصول على طول وعرض السجاد .
٢ ـ ضرب الطول فى العرض وتخزين الناتج .
٣ ـ كتابة الناتج .

فالخطوة الأولى تتطلب إدخالاً وتخزيناً ، والثانية تتطلب عمليات حسابية وتخزيناً ، والثالثة تتطلب إخراجاً للنتائج . ومن ثم لابد من إدخال المعلومات المكونة من الطول والعرض الفعليين :
٤٥ ، ٦٦ متراً مثلاً .

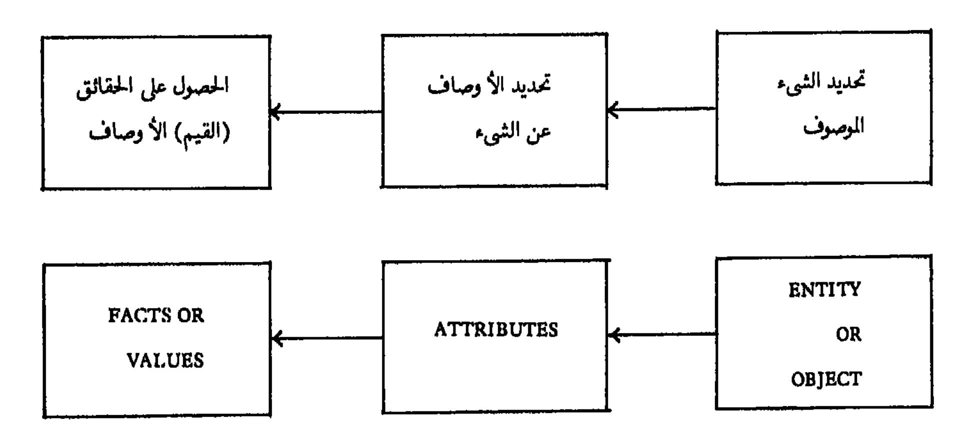
و يلخص الجدول التالى وظائف الحاسب الآلى :

الوظائف الأساسية	الوظائف الفرعية
١ ـــ العمليات الحسابية . ٢ ـــ العمليات المنطقية . **الوظائف المكملة :** ١ ـــ عمليات الإدخال ٢ ـــ عمليات التخزين . ٣ ـــ عمليات الإخراج .	١ ـــ التجميع والاختصار . ٢ ـــ التبو يب والتصنيف . ٣ ـــ الفرز والدمج . ٤ ـــ الفهرسة والتقسيم .

البيانات والمعلومات والتعليمات :

تعكس هذه الكلمات الثلاث الوظائف التى يقوم بها الحاسب الآلى ، وسنبين معنى كل منها وعلاقتها مع هذه الوظائف وعلاقة بعضها ببعض .

البيانات : من الممكن تعريف البيانات بأنها : «حقائق (قيم) تعكس أوصافاً (خصائص) معينة عن شى محدد» ؛ لذلك يكون التسلسل المنطقى للوصول إلى البيانات هو :



والشيء الموصوف Entity or object يمكن أن يكون مادياً ملموساً أو معنوياً ، والملموس إما أن يكون من الأحياء أو من الجماد ، ومن الأمثلة على كل نوع :

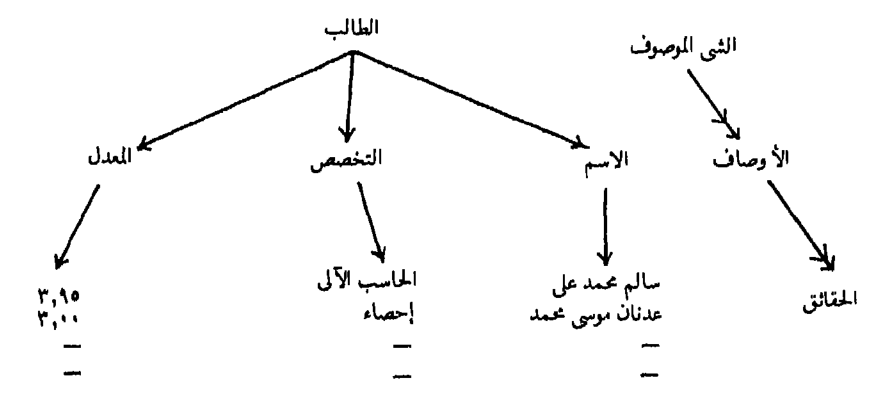
الشيء الملموس فى الأحياء : الموظف ، الطالب ، المدرس ، الشرطى ، الحيوان الأليف ، الدابة .

الشيء الملموس من الجماد : أسهم الشركة ، البضاعة ، قطعة الغيار ، غرفة الدراسة ، السيارة .

الشيء المعنوى : الشركة ، الطقس .

وأوصاف الأشياء تحدد بوظائفها ، فهى إما أن تكون أوصافاً لها علاقة بالعمليات الحسابية ، كطول وعرض ووزن ودرجة حرارة ، وعمر وتاريخ البلاد ، أو لها علاقة بالعمليات غير الحسابية ، كاسم الموظف وعنوانه ، ونوع الشركة ، ووصف قطعة الغيار . **والحقائق تتكون من أنواع الأحرف الثلاثة :** الرقمية (٠ – ٩) ، الأبجدية (أ – ى) ، والخاصة (مثل/!؟،،) . فالأوصاف الحسابية يجب أن تتكون من أرقام فقط ، وغير الحسابية يمكن أن تتكون من أى من أنواع الأحرف الثلاثة أو خليط من اثنين أو أكثر . فمثلاً يكون العمر «٢٥» سنة ، والاسم «سلمان على» والعنوان «ص . ب . ١١١ ، الرياض ١١١٤٥» ورقم القطعة «ص ت ج/ ٢١٦٥»

ويمكن توضيح العلاقة بين العناصر الثلاثة بالشكل الهرمى التالى :



و يلاحظ أحادية الشيء الموصوف ، وتعدد الأ وصاف للشيء المحدد ، وتعدد الحقائق لكل وصف .

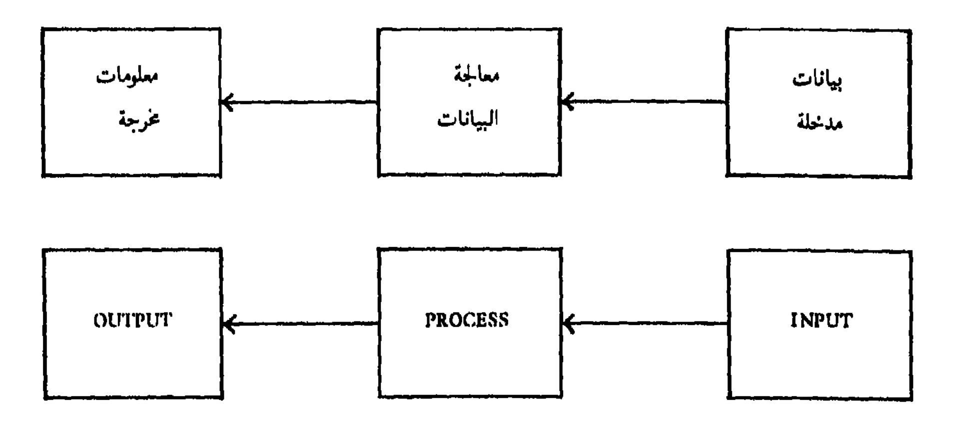
المعلومات : ويمكن تعريف المعلومات بأنها : البيانات التى تمت معالجتها حسابياً و/ أو منطقياً ، ومن ثم عرضها على المستفيد بأسلوب يحقق الاستفادة المرجوة منها . فقد سبق طباعة الشيك مثلاً عمليات حسابية للوصول إلى صافى الراتب من مجمله ، وعمليات منطقية لفرز الشيكات حسب رقم الموظف ، ومن ثم يتم اختيار الورق المناسب كوسيلة لصرف المبلغ من المصرف ، وتطبع عليه العبارات المناسبة ، مثل اسم المؤسسة وغيرها من العبارات . فمثلا تطبع عبارات «معهد الإدارة العامة» و «ادفعوا لأمر» و «المدير المالى» و «أمين الصندوق» ، وأخيراً تتم طباعة الشيكات فى نهاية كل شهر لتوافق الدورة المحاسبية ، وتعطى للموظف ليقوم بتسلم أو إيداع المبلغ .

و يلاحظ أن المعلومات تتألف من :

	بيانات مدخلة		بيانات مخرجة		العبارات
المعلومات =	مجمل الراتب	+	صافى الراتب	+	معهد الإدارة العامة
	العلاوات		مجموع العلاوات		ص. ب ٢٠٥
	الخصومات		مجموع الخصومات		ادفعوا الأمر

و بالإضافة إلى الطباعة على الورق ، يمكن استخدام الشاشة التلفازية (CRT) كوسيلة لعرض المعلومات .

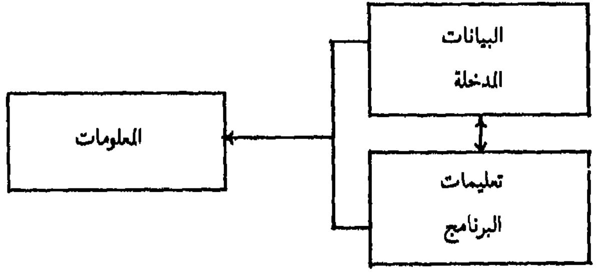
التعليمات : يمكن تعريف التعليمات بأنها : الخطوات التى يجب اتباعها لتحو يل البيانات المدخلة إلى معلومات وفق الهدف المحدد ، وهذه الخطوات هى نفسها التى يقوم بها الإنسان للتوصل إلى المعلومات المطلوبة ، ولكنها قد تكون أكثر تحديداً وتفصيلاً عند توصيفها للحاسب . وتتبع هذه الخطوات الإطار العام التالى :



ولتحقيق هدف معين قد تتعدد وتتداخل البيانات المدخلة والمعالجات والمعلومات المخرجة . فالبيانات المدخلة قد تكون حقائق تعكس أشياء متعددة ، والمعالجات قد تكون حسابية و/ أو منطقية ، والمخرجات من معالجة معينة قد تؤخذ كلياً أو جزئياً كمدخلات لمعالجة أخرى ... وهكذا .

ولما كان التخاطب مع الحاسب بوضعه الحالى لايمكن أن يتم بواسطة اللغة الإنسانية ، فقد طورت لغات ذات تعليمات محددة لتسهيل عملية التخاطب والاتصال مع الحاسب ، وتتم برمجة الخطوات وإيصالها إلى الحاسب باستخدام إحدى هذه اللغات . و يطلق على الخطوات المترجمة بواسطة تعليمات لغة البرمجة ، والتى تحقق هدفاً محدداً ، اسم البرنامج : (Program) .

و يعكس الرسم التالى العلاقة بين العناصر الثلاثة :



حيث يتم التفاعل والاتصال بين البرنامج والبيانات لإنتاج المعلومات .

الهيكل الهرمي للبيانات (Hiecarchical Structure of Data) :

حيث إن التعامل مع البيانات يشكل النشاط الرئيسي في تحقيق الهدف من استخدام الحاسب، كان لابد من وضع قواعد لتسهيل عملية تخزين واسترجاع ومعالجة هذه البيانات .

يطلق اسم «الحقل» على الأوصاف (الخصائص) التابعة للشيء المحدد، فالعمر والاسم يسميان حقلاً .

ويطلق اسم «السجل» على الحقائق (القيم) التابعة لعنصر واحد من الشيء الموصوف . فالحقائق التالية التابعة لطالب معين تعتبر سجلاً :

سالم محمد على الحاسب الآلى ٣,٩٥

ويطلق اسم «الملف» على جميع السجلات التابعة للشيء الموصوف .

وبناء على ذلك يكون الهيكل الهرمي للبيانات على الشكل التالى :

الملف	(File)
السجل	(Record)
الحقل	(Field)

حيث يوجد لكل ملف أكثر من سجل، ولكل سجل أكثر من حقل . ويتم تخزين البيانات عن طريق الحصول على قيم الحقول المتعددة التابعة لسجل معين، ومن ثم تخزينها إذا تطلب الأمر فى ملف واحد، ولا تعالج السجلات المتتابعة دون تخزينها فى الملف . وإذا تم تخزين الخصائص (القيم) ضمن ملف وأريد استرجاعها، فمن الممكن استرجاع الملف كاملاً بسجلاته جميعها، أويمكن اختيار مجموعة سجلات من الملف، كاختيار الطلبة المتفوقين الذين تزيد معدلاتهم على ٣,٥٠، أو فرز وتصنيف السجلات حسب التخصصات مثلاً، أويمكن استرجاع قيم حقول معينة تابعة لسجل وملف محددين، للحصول على معدل طالب معين، مثلاً، من عدة علامات معطاة . وسنتعرض

هنا بإيجاز لأساليب استرجاع البيانات من الملفات ، حيث ستتضح كيفية تطبيق هذه الأساليب فى الفصل الثالث عشر الخاص بمعالجة الملفات .

أساليب استرجاع البيانات File Access Methods :

تستند أساليب استرجاع البيانات إلى مفهومين أساسيين :
الأول : الاسترجاع المتتالى Sequential ، والثانى : الاسترجاع العشوائى Random .

الاسترجاع المتتالى : إن ترتيب السجلات فى ملف معين حسب هذا الأسلوب يسمح باسترجاع هذه السجلات بطريقة متتالية ، وذلك يشبه التسجيل على الأشرطة السمعية والبصرية . وعند محاولة استرجاع هذه البيانات نواجه بحالتين :

١ ـ استرجاع السجلات كلها فى الملف المعين ، كاسترجاع ملف الموظفين لأجل طباعة الشيكات ، أو استرجاع ملف الدارسين لأجل طباعة كشوف العلامات ، وذلك سجلاً سجلاً ودون تخطى أى منها .

٢ ـ استرجاع سجل معين ، لموظف محدد أو دارس معين ، ولا يتم ذلك إلا بعد استرجاع جميع السجلات السابقة على السجل المطلوب . فمثلاً : إذا كان اسم الدارس الأخير «سالم» ، فيجب استرجاع كل سجل سابق على سجل الدارس والتأكد فى كل مرة من أن السجل المسترجع هو المطلوب أم لا ، إلى أن نتوصل إلى استرجاع السجل المطلوب للدارس المعين .

لذلك كان ترتيب السجلات فى سجل متتالٍ (متتابع) يستند إلى حقل معين فى ذلك السجل مثل رقم الموظف أو اسم الدارس ، وذلك فى أغلب الأحيان . وفى بعض الأحيان ، يكون تسلسل السجلات فى الملف وفق التسلسل الزمنى لوقوع تسجيل المعلومات على كل سجل .

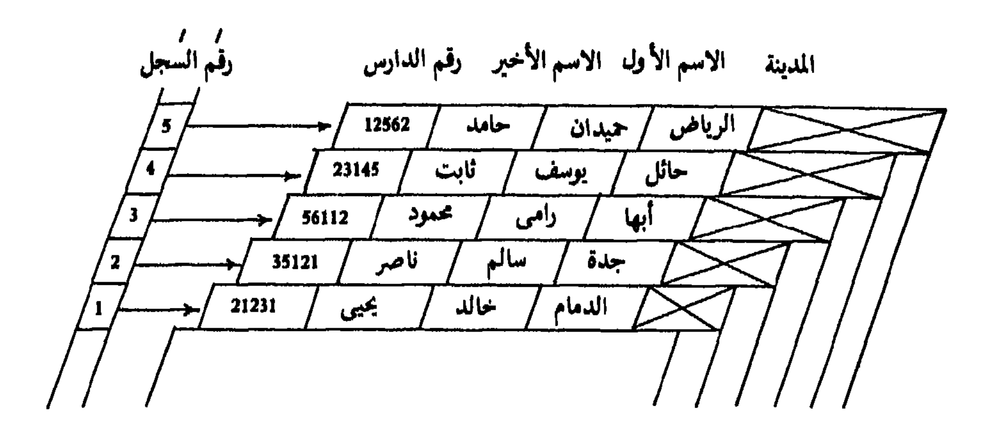
و يتبع هذا الأسلوب فى الحالات التى تتطلب استرجاع جميع سجلات الملف بصفة دورية، أسبوعياً، أو شهرياً، أو سنوياً، وذلك مثل تجهيز درجات الدارسين عقب كل دورة أو فصل، أو تحضير الشيكات للموظفين شهرياً، أو تجهيز الفواتير للكهرباء أو الهاتف شهرياً. وكذلك فى حالات استرجاع نسبة كبيرة من مجموع السجلات، كتلك التى تتبع منطقة جغرافية معينة أو نوعاً معينا من الخدمات أو البضائع، بحيث يكون عدد هذه التصنيفات فى الملف قليلا.

الاسترجاع العشوائى : و يسمح هذا الأسلوب باسترجاع أى سجل من السجلات دون الحاجة إلى استرجاع ماسبقه. مع العلم بأن هذا الترتيب يسمح باسترجاع البيانات بطريقة متتالية لنفس الملف. و يتم تحديد ترتيب السجلات واسترجاعها وفق إحدى طريقتين :

١ ـــ استخدام رقم كل سجل لايكون من الحقول التابعة له والمخزنة فى كل سجل، حيث يربط الحاسب ـــ عن طريق البرنامج ـــ بين كل سجل ورقمه. ولاسترجاع السجل المطلوب لابد من تحديد رقمه أولاً، ومن ثم استرجاع الحقول التابعة له.

٢ ـــ استخدام أحد الحقول التابعة للسجل، لتحديد كل واحد منها بطريقة فريدة، كما فى ترتيب الملف العشوائى. ومن الأمثلة عن الحقل الذى يمكن استخدامه ـــ والذى يطلق عليه الحقل المفتاح Key Field ـــ اسم الدارس أو الموظف (أو رقمه)، ورقم الفاتورة ورقم الطلبية. ولا يشترط ترتيب السجل تسلسلياً حسب قيمة هذا الحقل كما هو الحال فى الملف التتابعى.

و يشرح الشكل التالى الأساليب المذكورة سابقاً، والذى يفترض وجود خمسة سجلات فى ملف الدارسين، يحوى كل ملف الحقول التالية : رقم الدارس، اسمه، والمدينة التى جاء منها :



١ ـ الأسلوب التتابعى : تم ترتيب السجلات وفق الاسم الأخير للدارس ، لذلك فلاسترجاع ملف الدارس رامى محمود ، لابد من استرجاع ملفى الدارسين : حامد وثابت ، قبل أن نتمكن من الوصول إلى الملف المطلوب .

٢ ـ الأسلوب العشوائى الذى يستند إلى حقل مخزن ضمن السجل : فى هذه الحالة تم تحديد كل سجل باستخدام رقم الدارس غير المكرر، فإذا أردنا استرجاع نفس السجل ، نخبر الحاسب بأننا نرغب فى استرجاع حقول الدراس رقم 56112 ، دون الحاجة إلى استرجاع أية سجلات أخرى قبله .

٣ ـ الأسلوب العشوائى المستند إلى رقم السجل غير المحفوظ ضمن حقول السجل نفسه : فعند استرجاع نفس السجل ، يتم إعلام الحاسب برقم السجل المطلوب «3» فى مثل هذه الحالة .

عوامل المفاضلة بين أساليب استرجاع البيانات : نوجز هنا بعض العوامل التى تؤثر فى استخدام هذه الأساليب :

١ ـ **التكلفة :** إن تكلفة الأجهزة المناسبة لتخزين السجلات بشكل متتابع هى أقل من تكلفة الأجهزة المطلوبة لتخزين السجلات بأسلوب عشوائى .

٢ ـ **طبيعة استرجاع البيانات :** إن استرجاع السجل كله أو جزء كبير منه فى كل مرة يستخدم فيها السجل، يستدعى استخدام الملف التتابعى . أما إذا كانت الحالة تتطلب استرجاع سجل محدد أو مجموعة قليلة من السجلات (تقريباً ١٠٪) فى كل مرة يستخدم فيها السجل، فإن ذلك يتطلب استخدام الأسلوب العشوائى.

٣ ـ **الاعتمادية :** تعتبر وسائط التخزين العشوائى أكثر اعتمادية من وسائط التخزين المتتالى .

٤ ـ **الهدف من التخزين :** تستخدم أجهزة التخزين التتابعى فى العادة لحفظ نسخ من السجلات لأغراض المحافظة عليها Back-up Copics ، وكذلك فى حالات التخزين التى تتطلب الحفاظ على معلومات تاريخية لفترة طويلة، ٣ ـ ١٠ سنوات مثلاً، لأغراض حكومية أو أمنية . فى حين تستخدم أجهزة التخزين العشوائى لتخزين المعلومات المستخدمة باستمرار .

٥ ـ **السرعة :** إن أجهزة الاسترجاع العشوائى (المباشر) أسرع من أجهزة التخزين المتتالى فى الوصول إلى الحقول التابعة للسجلات .

٦ ـ **قدرات الاستفهام Inquiry المباشر :** تمتلك وسائط التخزين المباشر العشوائى إمكانية الربط المباشر المستمر مع الحاسب، بحيث تسمح بالاسترجاع اللحظى الآتى للمعلومات والبيانات، فى حين تخزن فى معظم الحالات، وسائط التخزين المتتالى بشكل منفصل ومستقل عن الحاسب، وتحمّل فيما بعد عند الحاجة على الأجهزة الخاصة بها .

٧ ـ **السعة التخزينية :** إن السعة التخزينية لأجهزة الوصول المباشر العشوائى هى أكبر بكثير من تلك السعة الخاصة بأجهزة التخزين المتتالى .

أجهزة التخزين : تنقسم أجهزة التخزين إلى نوعين رئيسيين :

١ ــ وسائط التخزين التى يتم التخزين المباشر للبيانات عليها ، وذلك كالقرص الرقيق Floppy Diskette أو مجموعة الأقراص Disk Pack ، حيث يستخدم الأول للحاسبات الشخصية الصغيرة ، والأخير للحاسبات الكبيرة . وهذه الوسائط تسمح بالوصول العشوائى للسجلات . أما الأشرطة المغناطيسية Tape التى تشبه تلك الخاصة بأجهزة التسجيل المصغرة Cassette tape أو الأجهزة الكبيرة Tape ، فتستخدم الأولى مع الحاسبات الشخصية ، و الثانية مع الحاسبات الكبيرة . وهذه الأشرطة تستخدم عند تطبيق أساليب الاسترجاع المتتالى للسجلات .

٢ ــ وحدة القراءة والكتابة ، والتى تستخدم لاسترجاع البيانات المخزنة على وسائط التخزين ، و يطلق وحدة القرص Disk Unit على الجهاز الخاص بالاسترجاع العشوائى ، ووحدة الشريط Tape Unit على الجهاز الخاص بالاسترجاع المتتالى .

انظر الشكل التالى (ص ٤٤) ، الذى يوضح أجزاء أجهزة الاسترجاع العشوائى والمتتالى .

نظرية النظام (Systems Theory) وتطبيقها مع الحاسب الآلى :

تعتبر نظرية النظام إحدى الوسائل المهمة التى تساعد فى فهم كيفية عمل وتفاعل مجموعة الأجزاء والمكونات ، التى تحقق هدفاً معيناً فى كثير من المجالات . هناك النظام المالى والنظام الاقتصادى والنظام التربوى . وبدلاً من الخوض فى تفصيلات أسس وقواعد نظرية النظام ، سيتم تقديم إطار مبسط يفى بالغرض من هذا الجزء من الكتاب .

مقدمة عن نظرية النظام :

يمكن تعريف النظام بأنه «مجموعة من الأجزاء (المكونات) ذات وظائف محددة تتفاعل فيما بينها لتحقيق هدف محدد» ، و بناء على ذلك يمكن تحديد معالم نظرية النظام كالتالى :



شكل ١ ــ ١ : أنواع أجهزة التخزين العشوائى والمتتالى

١ ــ تحديد الهدف من النظام . وهذا الهدف يكون عاماً تلحق به أهداف فرعية تفصيلية ، حسب التعقيد فى وظائف الأجزاء وتفاعلها .

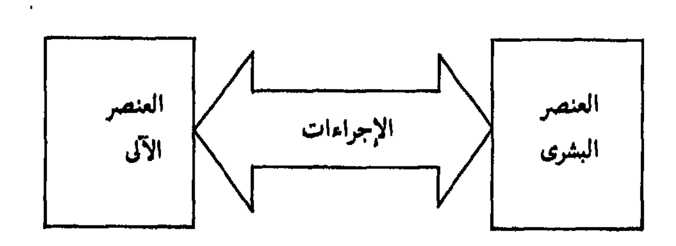
٢ ــ تحديد الوظائف التى يجب أن تتم لتحقيق الهدف العام والأهداف الفرعية .

٣ ــ تحديد المكونات (الأجزاء) التى ستقوم بهذه الوظائف . و يلاحظ أن كل جزء يجب أن يقوم بمهمة محددة تؤدى فى النهاية إلى تحقيق الهدف العام .

٤ ــ تحديد أسس التفاعل بين هذه الأجزاء حتى يتم تحقيق الهدف العام والأهداف الفرعية ، بطريقة متناسقة ومتجانسة وتمنع أى اضطراب أو خلل فى تأدية الوظائف .

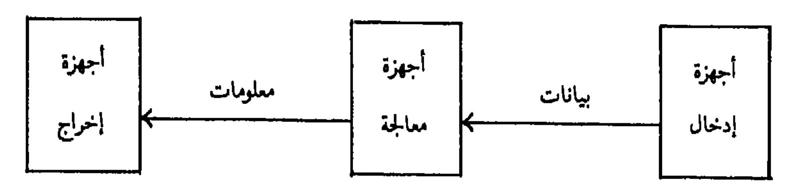
عناصر (مكونات) نظام الحاسب الآلى :

إن الهدف العام من نظام الحاسب الآلى هو تحقيق متطلبات المستفيدين عن طريق الاستخدام الأمثل لأجهزة الحاسب . لذلك فإن نظام الحاسب الآلى يعتمد بشكل رئيسى على عملية الاتصال مع العنصر البشرى (الإنسان) والعنصر الآلى (أجهزة الحاسب)، و يتم تحديد أسس الاتصال عن طريق الإجراءات التفصيلية . و يعكس الشكل التالى الإطار العام لنظام الحاسب الآلى :



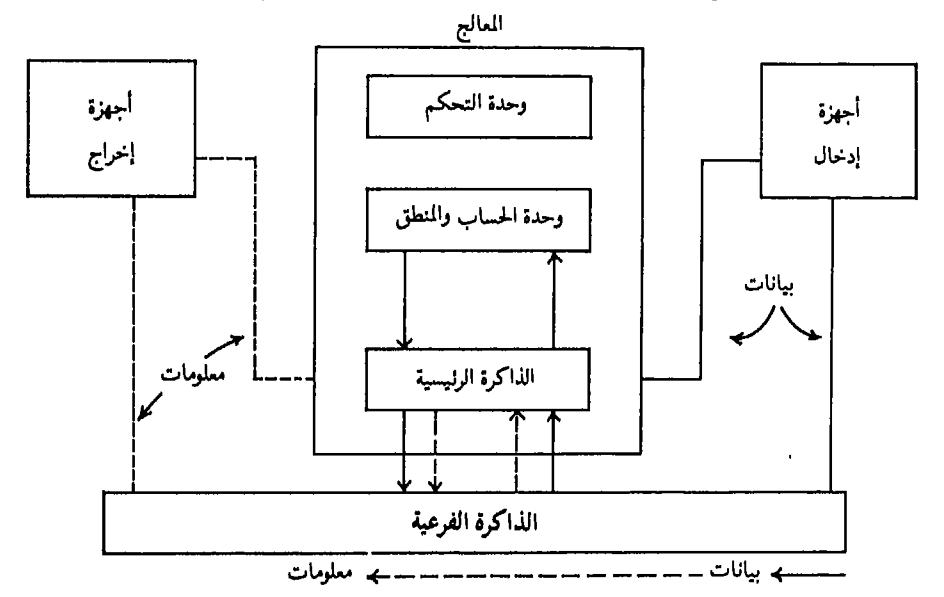
وسيتم التركيز هنا على العنصر الآلى وذلك لأهميته فى فهم أسس البرمجة ، فى حين ستتم مناقشة العنصرين الآخرين بإيجاز، وذلك لعلاقتهما بموضوع التحليل والتصميم .

العنصر الآلى : لقد تمت مناقشة وظائف الحاسب فى الجزء الأول من هذا الفصل . فالهدف من العنصر الآلى هو معالجة البيانات وإخراجها على شكل معلومات ، لذلك لابد من وجود أجهزة إدخال ومعالجة وإخراج . أما تفاعل هذه الأجزاء فيمكن وصفه بالرسم التالى :



وإذا أضيفت وظائف التخزين والمعالجات الحسابية والمنطقية وضرورة وجود جزء يتحكم فى التفاعل بين هذه الأجزاء جميعاً ، يمكن وصف النظام الفرعى للعنصر الآلى بالشكل التالى . و يطلق تعبير الأجهزة المتعلقة Peripheral devices على أجهزة الإدخال والإخراج والذاكرة الفرعية . هذا وسيتعرض الجزء الأخير من هذا الفصل لأنواع هذه الأجهزة مع الذاكرة الرئيسية .

شكل ١ ــ ٢ : مكونات الحاسب الآلى وتفاعلها بعضها مع بعض



وبعد تحديد الهدف العام ووظائف ومكونات الأجزاء لنظام عنصر الحاسب الآلى (الأجهزة)، يمكن عرض بعض قواعد التفاعل بين الأجزاء :

١ ـــ لايمكن لأى جزء أن يقوم بأية مهمة دون أمر مباشر من وحدة التحكم . وهذا يتم داخلياً وليس له علاقة بعملية برمجة التعليمات .

٢ ـــ للحصول على المعلومات لابد من أن تتم المعالجات المطلوبة على البيانات المدخلة . وهذا يتطلب أن تكون البيانات المدخلة صحيحة، وأن تكون خطوات المعالجة متكاملة وواضحة، لذلك فإن أى خطأ فى المعلومات يكون سببه إما خطأ فى البيانات، أو فى ترجمة خطوات المعالجة إلى تعليمات فى البرنامج، أو عدم وجود البيانات بالكلية، أو معالجة البيانات التابعة لمعالجة أخرى خاطئة .

٣ ـــ يمكن تخزين البيانات إما فى الذاكرة الرئيسية، وذلك إذا أريد معالجتها آنياً (لحظياً)، أو فى الذاكرة الفرعية إذا أمكن الانتظار لحين وجود مكان شاغر، فى الذاكرة الرئيسية . ولما كانت الذاكرة الرئيسية تعتبر الذاكرة النشطة والمستعدة دائما لاستقبال البيانات، فمن الأفضل تخفيف العبء عنها باستخدام الذاكرة الفرعية إلا فى الحالات التى تتطلب سرعة ومعالجة آنية لكل سجل على حدة، كحجوزات الطيران مثلا . أما إذا كان المطلوب معالجة مجموعة سجلات بعضها مع بعض ـــ عند طباعة الفواتير مثلا ـــ فيمكن استخدام الذاكرة الفرعية بدلا من الرئيسية .

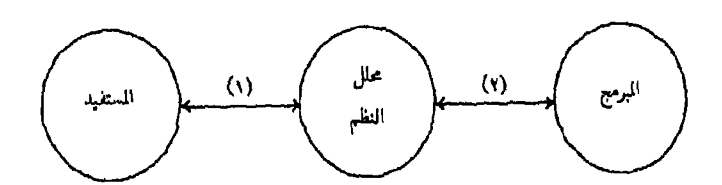
٤ ـــ إنه، حتى تتم معالجة البيانات، لابد من وجودها فى الذاكرة الرئيسية . ولذلك فهناك عملية إضافية عند تخزين البيانات فى الذاكرة الفرعية، يجب الانتباه إليها، وهى ضرورة نقل البيانات من الذاكرة الفرعية إلى الرئيسية حين الحاجة إلى معالجتها .

٥ ــ إن وحدة الحساب والمنطق تستقبل وترسل من الذاكرة الرئيسية دون غيرها ، البيانات فقط التى تتطلب معالجة حسابية أو منطقية .

ولابد من التنويه هنا ، وذلك عند التحدث عن الأجهزة (HARDWARE) ، بأن تشغيل هذه الأجهزة والاستفادة من قدرات وخصائص الحاسب يتمان عن طريق البرمجيات (SOFTWARE) بأنواعها المختلفة ، وسيتم التحدث بمزيد من التفصيل عن هذا الجزء عند مناقشة عنصر الإجراءات .

العنصر البشرى : يدخل فى هذا العنصر كل من يستخدم الحاسب الآلى ، سواء الموظفون فى مركز الحاسب أو الموظفون فى الجهات المستفيدة من قدرات الحاسب ، كالدائرة المالية ودائرة شؤون الموظفين . وتكون مهمة الموظفين فى مركز الحاسب منصبة على التعرف على رغبات ومتطلبات المستفيدين ، وصياغتها بطريقة يسهل معها تحقيقها بواسطة استخدام الحاسب ، أو يدوياً ، أو باستعمال كلتا الوسيلتين .

و يهمنا هنا الاتصال الذى يتم بين ثلاث فئات من المستخدمين ، كما يبينه الشكل التالى ، حيث يدل الرقم على التسلسل المنطقى لهذه الاتصالات :



يقوم محلل النظم ، أثناء و بعد الاتصال مع المستفيد ، بصياغة متطلبات المستفيدين بطريقة تسهل أداء المبرمج لعمله .

و يسلم محلل النظم إلى المبرمج ما له علاقة بكتابة البرامج وفق إحدى لغات البرمجة ، وتتضمن التالى :

١ ــ تحديد المدخلات للبرنامج .

٢ ــ وصف الإجراءات الحسابية والمنطقية بإحدى الوسائل المتاحة لمحلل النظم .

٣ ــ تحديد المخرجات وأسلوب وشكل (عرض) المعلومات .

هذا إذا كانت البرامج المطورة جزءاً من نظام متكامل يخدم فئات متعددة من المستفيدين . أما إذا كان البرنامج ذا هدف محدد ليس له علاقة ببرامج أخرى، فقد يقوم المبرمج بتحديد البيانات المدخلة والمعالجات والمخرجات والمعلومات . هذا حال التمارين فى أى كتاب فى البرمجة إلا فى بعض الأمثلة المعقدة، وهى قليلة .

وفى هذا الكتاب، سيتم تحديد أسلوب وشكل عرض المعلومات لبعض التمارين، و يطلب ذلك من المبرمج فى البعض الآخر؛ وذلك لإعطائه فرصة الممارسة التى ستفيده عند كتابة تطبيقات خاصة به، أو عند عمله فى جهة قد تتطلب الا تصال مع المستفيدين أو محلل النظم .

عنصر الإجراءات : تتضمن الإجراءات وصفاً للخطوات والعمليات والأنشطة التى تصاحب استغلال قدرات الحاسب من قبل المستخدمين .

وتقسم الإجراءات إلى قسمين : تلك التى تخص المستفيد، وتلك التى تخص الموظفين فى مركز الحاسب . وسيتم إعطاء صورة إجمالية عن هذه الإجراءات مع التعرض لأهميتها .

وإجراءات المستفيدين لها علاقة باستخدام الحاسب وتجميع البيانات وتسلم وتوزيع المعلومات، وأهمها هو دليل المستفيد الخاص باستخدام البرامج التى طورت لتحقيق متطلبات المستفيدين، مثل هؤلاء الذين يعملون فى الدوائر المالية والإدارية وشؤون الموظفين .

وتقسم إجراءات المستخدمين فى مركز الحاسب الآلى إلى تلك التى لها علاقة بالنظم التشغيلية (برمجيات النظم Systems Software) وهى غالباً ما تكون مكتوبة من

قبل الشركات المصنعة للحاسب . وتلك التى لها علاقة بالنظم التطبيقية (برمجيات التطبيقات Applications Software) ، وهى التى طورت من قبل المحللين والمبرمجين ، سواء داخل المؤسسة أو عن طريق الاستئجار أو الشراء . وموضوع هذا الكتاب له علاقة بالقسم الأخير .

وتتبع عملية توثيق إجراءات النظم التطبيقية، نشاطات ووسائل متعددة تؤدى إلى تحضير أدلة وتقارير ذات أهداف محددة . فمن الأدلة ماله علاقة بوصف إجراءات التحليل والبرمجة ، أو بوصف إجراءات التشغيل لتطبيق معين على الحاسب . ومن التقارير ما يقدم على مراحل مختلفة إلى المستفيدين ، وذلك لإطلاعهم على التقدم فى تطوير تطبيق ما ، وإعطائهم معلومات تساعدهم فى اتخاذ قرارات حول جدوى التطبيق المطور .

وما يهمنا فى هذا الكتاب هو التوثيق المبسط للبرامج المكتوبة من تخصيص البيانات المدخلة ، ووصف لخطوات المعالجة باللغة العربية وتخصيص المخرجات وأشكال عرض المعلومات ، وذلك للتدليل على أهمية التوثيق للبرامج .

أقسام الحاسبات الآلية حسب حجمها :

كان اتجاه حجم الحاسبات خلال التطور التاريخى لها هو من الكبير إلى الصغير، وهذا بخلاف تصغير حجم الحاسب التابع لفئة معينة . وقد نجم ذلك كله عن الأساليب الحديثة فى تصنيع أجزاء الحاسبات الآلية ، و بشكل خاص جزء المعالج ، وعن استعمال التيار الكهربائى فى تصميم الحاسب ، عوضاً عن الأساليب الإلكتروميكانيكية أو الإلكترومغناطيسية . ولا يهمنا هنا التفصيل فى هذا الموضوع ، ولكن يهمنا فقط التنبيه إلى أن جميع أعمال الحاسب الآلى الحالى تتم عن طريق إرسال تيارات كهربائية عبر أسلاك دون أن يتم تحريك أى جزء أو قطعة منها .

ومن الممكن تقسيم الحاسبات حسب أحجامها إلى الفئات التالية مع التعرض لأهم العوامل التى تتطلب استخدامها :

١ ــ الحاسبات ذات الحجم الضخم : (Super-Computers) وتستخدم هذه الحاسبات فى التطبيقات التى تتطلب سرعة ودقة متناهيتين فى معالجة أرقام كبيرة، مثل الفضاء واكتشاف وتصنيع البترول .

٢ ــ الحاسبات ذات الحجم الكبير : (Main-Frame) وتستخدم هذه الحاسبات فى الشركات الكبيرة التى تتطلب تخزين البيانات الكثيرة مركزياً مع تحقيق سرعة ودقة مناسبة، مثل المؤسسات المالية الكبيرة و بعض تطبيقات المؤسسات الحكومية الكبيرة، مثل الداخلية وديوان الخدمة المدنية .

٣ ــ الحاسبات ذات الحجم الوسط (Mini-Computers) : تستخدم هذه الحاسبات فى الشركات متوسطة الحجم والتى تتطلب مركزية التخزين للمعلومات، أو فى الشركات الكبيرة حيث يتم تخزين لا مركزى لجزء من البيانات والتى تتم معالجتها محلياً، أو عند استعمالها لتطبيق خاص مثل التصنيع والتصميم بمساعدة الحاسب (CADQAM) لتطبيقات المعاهد والمؤسسات التعليمية والتدريبية .

٤ ــ الحاسبات ذات الحجم الصغير (small Computers) : تستخدم هذه الحاسبات فى الشركات الصغيرة، والتى لاتحتاج إلى طاقة تخزينية عالية حيث كمية البيانات محدودة إلى حد ما، ولا إلى التنوع والكثرة فى أجهزة الإدخال والإخراج، أو فى الشركات الأخرى التى لاتحتاج إلى مركزية فى تخزين ومعالجة البيانات .

٥ ــ الحاسبات الشخصية ذات الحجم المصغر (Personal Microcomputers)
وهذه الحاسبات إما أن تكون عادية ثابتة توضع فوق المكتب، أو متنقلة يمكن حملها كحقيبة أثناء السفر، وتستخدم للأغراض الشخصية سواء فى الأعمال الخاصة

بنشاطات الشركات التابعة لشخص واحد أو لأشخاص قليلين، أو فى أعمال المديرين التى تتطلب معالجات ذات صفة خاصة نابعة من طبيعة وظائفهم كالقيام بالتحليلات الإحصائية، أو متابعة النشاطات المنزلية لأفراد الأسرة، كمراقبة المصروفات والإيرادات .

وتحدد هذه الحاسبات عدد المستخدمين بمستخدم واحد فى معظم الأحيان (Single User)، مع أن التقدم فى تصنيع أجهزة المعالجة أدى إلى إتاحة استخدامها من قبل مستخدمين متعددين (Multi-Users)، مع أن عدد المستخدمين قد لا يصل إلى ذلك العدد المتاح على الحاسبات ذات الحجم الصغير .

وقد تستخدم هذه الحاسبات إما مستقلة بذاتها أو متصلة مع حاسب من النوع ذى الحجم الوسط أو الكبير، وتتمتع هذه الحاسبات بسهولة توصيلها إلى مصدر للتيار الكهربائى وقلة صيانتها .

٦ ـ الحاسبات المنزلية ذات الحجم الصغير (Home Micro-Computers) وهى شبيهة بالحاسبات الشخصية إلا أن قدراتها أقل منها . وهى غالباً ما تستعمل فى الترفيه والتسلية عن طريق استخدام الألعاب، مع إمكانية استخدامها لأغراض متابعة مصروفات وإيرادات الأسرة، وهى من ذات المستخدم الواحد (Single-User) .

ومن الممكن تبيان الفروق بين هذه الأنواع من الحاسبات من خلال مناقشة بعض الخصائص :

١ ـ الحجم : تتراوح أحجام الحاسبات من تلك التى تشبه حقيبة سفر إلى تلك التى تملأ حجرة كبيرة الحجم، هذا دون اعتبار لأجهزة الإدخال والإخراج التى قد تكون موزعة فى أماكن متباعدة وتأخذ حيزاً كبيراً .

٢ ـ حجم الذاكرة الرئيسية : تتراوح حجم الذاكرة الرئيسية بين ٤ آلاف حرف (٤ ك ح) وما يزيد على ٣٢ مليون حرف (٣٢ م ح) .

٣ ــ حجم الذاكرة الفرعية : كما مر ذكره، تقسم الذاكرة الفرعية إلى ذاكرة وصول متتالٍ وذاكرة وصول عشوائى . وكذلك تختلف الوسائط المستخدمة لكل منهما حسب حجم الحاسب، فالحاسبات الشخصية المصغرة من جهة، والحاسبات الأخرى من جهة ثانية .

وتختلف سعة التخزين المتتالى ــ الأشرطة ــ حسب طولها .

أما سعة التخزين العشوائى ــ الأقراص ــ فتتفاوت حسب الوسائط المستخدمة . فالقرص الرقيق التقليدى تأتى سعته إما ٣٦٠ الف حرف أو ١,٢ مليون حرف، أما القرص الصّلب المصغر، والذى يستخدم مع جهاز IBM PS/2 ، فيمكن أن يكون إما ٧٢٠ الف حرف أو ١,٤٤ مليون حرف . هذا فيما يتعلق بالحاسبات الشخصية . أما فى الحاسبات الكبيرة، فتتفاوت سعة وسائط التخزين العشوائى لمجموعة الأقراص .

٤ ــ عدد وحدات الإدخال والإخراج : تعتبر الطرفيات النهائية المكونة من لوحة الأزرار والشاشة التلفازية أحد أجهزة الإدخال والإخراج الرئيسية على التوالى لكل الحاسبات . فى حين تعتبر الطابعات، بأنواعها المختلفة، إحدى الوسائل الشائعة للحصول على معلومات مكتوبة . وكذلك من الممكن كما مر ذكره، استخدام الذاكرة الفرعية كوسيلة إخراج وإدخال أيضا، وذلك إذا ما تم تخزين البيانات المخرجة عليها، ومن ثم أدخلت لمعالجتها .

و يتراوح عدد الوحدات المتعلقة بالحاسب (Peripheral Devices) بين وحدتين وما يزيد على ٥٠٠ وحدة . ففى الحاسبات الصغيرة ذات المستخدم الواحد يمكن إيصال طابعة ومسجل بالإضافة إلى لوحة الأزرار والشاشة .

أما فى الحاسبات الكبيرة، فيمكن إيصال عدد كبير من النهائيات الطرفية ووحدات إدخال الأشرطة ووحدات إدخال الأقراص والطابعات، وذلك حسب الحاجة إلى كل نوع منها .

٥ ــ **السرعة :** تقاس سرعة الحاسب فى القيام بالعمليات الحسابية والمنطقية بـ (١/ ١٠٠٠) من الثانية . فى حين تقاس سرعة استجابة الحاسب لطلب المستخدم ، من لوحة الأزرار إلى الشاشة مثلاً ، بالثوانى . وكلما زاد حجم الحاسب زادت السرعة فى إنجاز أعماله .

٦ ــ **أنواع وتعدد لغات البرمجة :** كلما زاد حجم الحاسب زاد معه عدد وأنواع لغات البرمجة . ففى الحاسبات ذات الحجم المصغر (Micro-Computers) ، تعتبر لغة بيسك ــ اللغة المشروحة فى هذا الكتاب ــ هى اللغة الرئيسية للحاسب ، حيث تستخدم للاتصال مع الذاكرة الرئيسية حال توصيل الحاسب إلى مصدر تيار كهربائى . ومن الممكن تغيير هذه اللغة إلى إحدى لغات قليلة أخرى ، مثل : كوبول ، وفورتران ، وباسكال . أما فى الحاسبات الأكبر حجماً ، فإن عدد وأنواع اللغات تكون كثيرة ، بالإضافة إلى إمكانية التخاطب والاتصال مع الحاسب باستخدام أكثر من لغة فى نفس الوقت .

تمارين

١ ــ عدّد خمسة من الأ وصاف المتعلقة بالدارس .

٢ ــ مؤسسة استشارية تقدم خدمات للزبائن عن طريق تقاريرعن طبيعة الصناعات المختلفة فى الشرق الأ وسط . حدد اثنين من الأشياء الموصوفة فى هذه المؤسسة . ومن ثم حدد خمسة من الأ وصاف لكليهما .

٣ ــ أعط خمس مجموعات من الحقائق (القيم) للأوصاف التالية المتعلقة بالشىء الموصوف .

الشىء الموصوف : الزبون .

الأ وصاف : رقم الزبون ، اسم الزبون ، عنوانه ، الرصيد المستحق ، تاريخ آخر دفعة ، تاريخ آخر طلبية .

٤ ــ ماهو المقصود بالتعابير التالية :

نظام ، أجهزة متعلقة ، برمجيات ، مبرمج ، محلل نظم ، مستفيدين ؟

٥ ــ اشرح باختصار وظائف الحاسب الرئيسية والفرعية .

٦ ــ ماهو الفرق بين البيانات والمعلومات والتعليمات؟

٧ ــ أعط مثالاً عن شىء موصوف مع تحديد ثلاثة أوصاف وإعطاء خمس قيم لهذه الأ وصاف .

٨ ــ ماهو الهيكل الهرمى للبيانات ، مع إعطاء مثال على ذلك؟

٩ ــ اشرح باختصار العناصر الثلاثة لاستخدام نظام الحاسب الآلى ، وما هى العلاقة بينها .

١٠ ـ اشرح باختصار مكونات الحاسب الآلى وتفاعلها بعضها مع بعض بواسطة الرسم؟

١١ ـ قارن بين أنواع الحاسبات حسب الخصائص التالية :

الحجم ، عدد المستخدمين ، نوع الذاكرة الفرعية ، عدد وحدات الإدخال والإخراج ، السرعة .

الفصل الثانى

مقدمة عن البرمجة التركيبية الحديثة

(Structured Programming)

كان الاهتمام الأكبر فى بداية الخمسينيات ينصب على نشاط البرمجة عند تطوير نظم التطبيقات . إذ أن نجاح المحاولات الأولى فى تحقيق الفائدة فى مجال السرعة والدقة فى معالجة البيانات الكثيرة، أدى إلى التسارع فى كتابة البرامج فى المجالات المختلفة الأخرى، وذلك حتى تعم الفائدة . ولم يكن هناك أى سبب يدعو إلى التفكير فى وضع أسس وقواعد لأساليب البرمجة .

أساليب البرمجة التقليدية :

فى هذا الجو المتفائل، تم تطوير كثير من البرامج باستخدام الأساليب التقليدية التى لاتخضع لأى أسس أو قواعد . ونظراً لكون البرمجة فناً وليست علماً ذا قواعد ثابتة، كانت البرامج المطورة فى ظل هذه الأساليب تتأثر إلى حد كبير بالخبرات الشخصية والفروق الفردية والميول الخاصة .

ومن خصائص أساليب البرمجة التقليدية :

١ ــ عدم وجود أية أسس وقواعد تتبع من قبل المبرمجين .

٢ ــ نظرة المبرمج إلى البرنامج وكأنه إنتاج خاص به، وأن تلازم المبرمج والبرنامج سيكون دون انقطاع .

٣ ــ عدم الأخذ بعين الاعتبار بأن البرامج المطورة ستخضع للتعديل فيما بعد .

٤ ـ كتابة البرامج دون تحليل مسبق ، مما أدى إلى عدم ترابط البرامج المختلفة المطورة على فترة من الزمن .

٥ ـ عدم وجود تعليمات فى لغة البرامج للتحكم فى الانتقال المبنى على نتيجة المقارنات المنطقية بين قيمتى حقلين ، أو أن استخدامها حال وجودها لم يكن يخضع لأى ترتيب واضح . لذلك كان المبرمج يقوم بالانتقال إلى أى تعليمة فى البرنامج سواء كانت سابقة أو لاحقة .

٦ ـ عدم تقسيم البرامج الكبيرة إلى أجزاء مترابطة بعضها مع بعض .

وقد اكتنفت البرامج التى طورت بهذه الأساليب صعوبات عديدة :

١ ـ صعوبة إمكانية تنقل البرامج من مؤسسة إلى أخرى إلا بصرف جهد ووقت كبيرين .

٢ ـ صعوبة إجراء تعديلات على هذه البرامج .

٣ ـ صعوبة فهم البرامج من قبل الأفراد الآخرين .

٤ ـ صعوبة التحقق من صحة المعلومات المخرجة .

٥ ـ صعوبة متابعة تأثير التغيير الذى تم فى جزء من البرنامج على الأجزاء الأخرى .

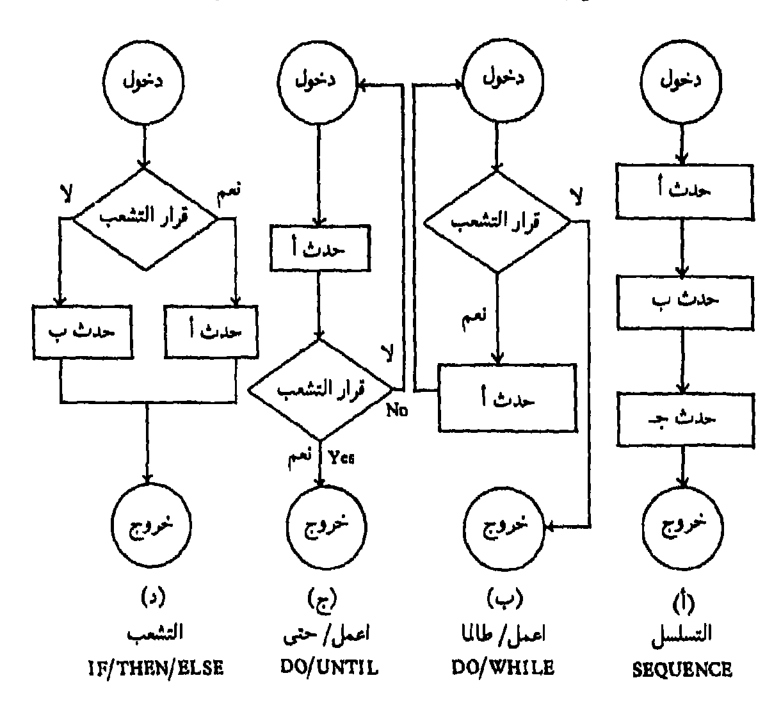
٦ ـ صعوبة معرفة إذا ماكان الخطأ الذى وقع فى جزء من البرنامج قد سبب وقوع أخطاء فى الأجزاء الأخرى .

وللتقليل من هذه الصعوبات ظهرت جهود ، بدأها الأستاذ دجكسترا (Edsger W. Dijkstra) العالم النمساوى ترمى إلى وضع أسس وقواعد للبرمجة الحديثة أطلق عليها «البرمجة التركيبية Structured Programming» .

أساليب البرمجة التركيبية الحديثة :

تحدد البرمجة التركيبية (ب ت) أربعة أنماط رئيسية، لا تباعها عند ترجمة الخطوات الخاصة بتحقيق الهدف من البرنامج إلى تعليمات إحدى لغات البرمجة . ويمكن توضيح هذه الأنماط الأساسية بالأشكال التالية :

شكل (٢ ـ ١) الأنماط الأربعة الأساسية للبرمجة التركيبية



وقبل مناقشة هذه الأساليب، لابد من توضيح الرموز المستعملة في الرسومات .

- تعبر عن بداية ونهاية النمط المحدد .
- تعبر عن معالجة حسابية بإحدى لغات البرمجة .
- تعبر عن اتخاذ قرار باختيار أحد المسارين استناداً إلى نتيجة المقارنة المنطقية بين قيمتي حقلين من البيانات .
- تعبر عن عمليات الإدخال و/ أو الإخراج .

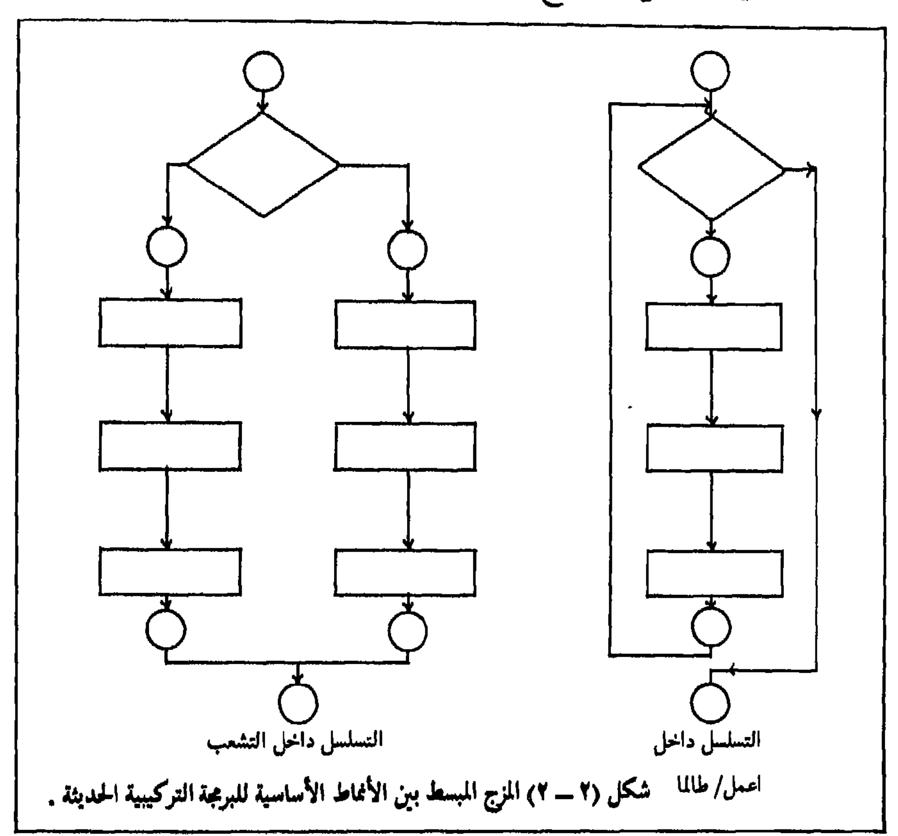
وفى الشكل (٢ ــ ١) يوضح الرسم (أ) التعليمات التى ستنفذ دون إجراء أية مقارنات تؤدى إلى التشعب، أى أن التعليمات ستنفذ فى تسلسل ثابت .

بينما الرسم (ب) يوضح أن نتيجة المقارنة المنطقية تحدد إذا ماكانت التعليمة ستنفذ أم لا .

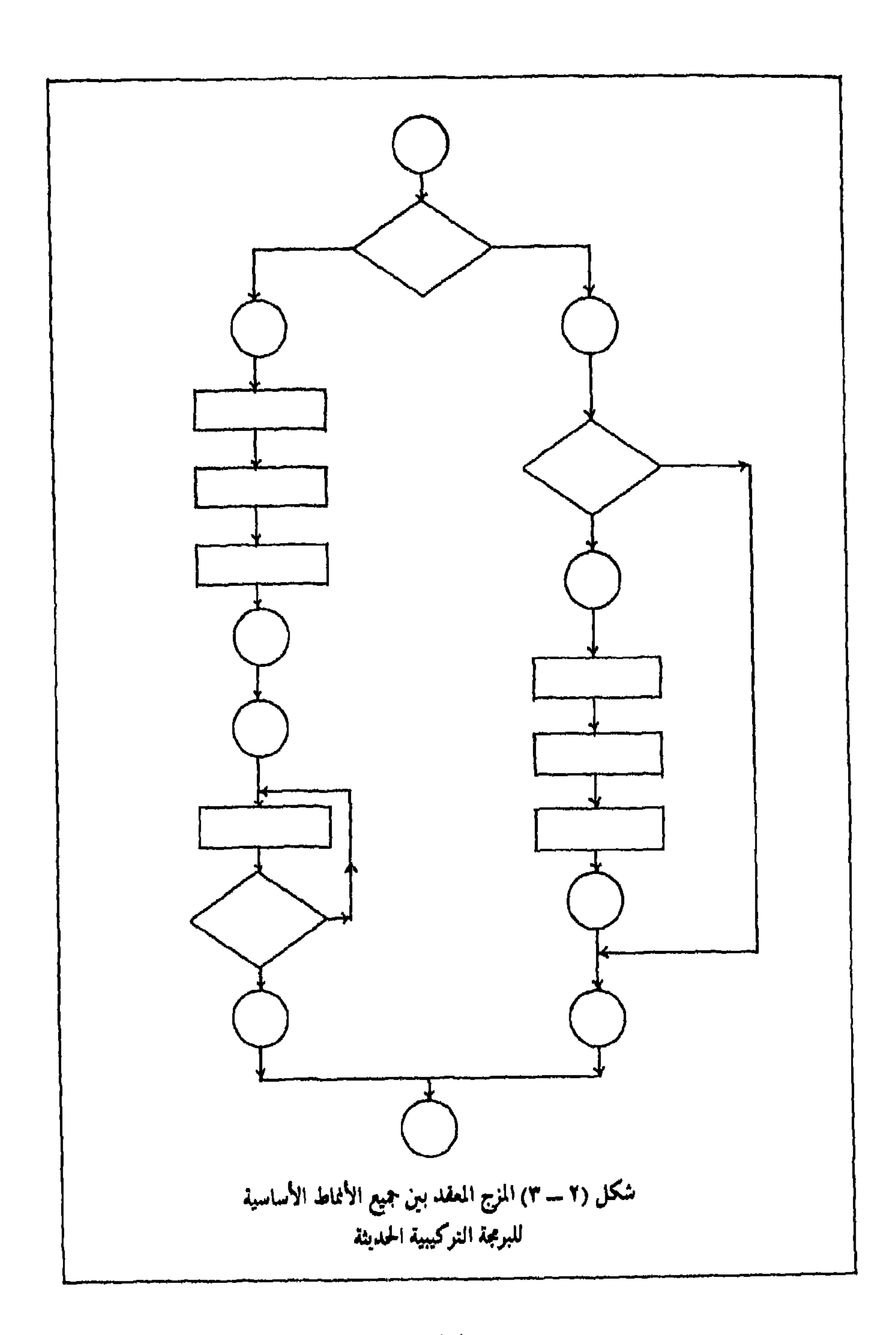
بينما الرسم (جـ) يوضح أن التعليمة ستنفذ أولاً مع استمرار تنفيذها استناداً إلى نتيجة المقارنة المنطقية .

بينما الرسم (د) يوضح اختيار تنفيذ إحدى التعليمتين بناء على نتيجة المقارنة المنطقية .

ومن الممكن أن تتطلب خطوات المعالجة المزج بين نمطين أو أكثر . وتعكس الأشكال التالية أمثلة من هذا النوع .



شكل (٢ ــ ٢) المزج المبسط بين الأنماط الأساسية للبرمجة التركيبية الحديثة .



شكل (٢ ـــ ٣) المزج المعقد بين جميع الأنماط الأساسية
للبرمجة التركيبية الحديثة

و يلاحظ من الأشكال السابقة، أن استخدام أساليب البرمجة الحديثة فن يتقن عن طريق الممارسة، و يتأثر بالإبداع الشخصي للمبرمج .

ولاستخدام هذه الأساليب تتبع القواعد التالية :

١ ــ تحديد بداية ونهاية كل نمط كجزء مستقل بذاته .

٢ ــ عدم تنفيذ تعليمة منفصلة داخل أى نمط محدد، وإنما يكون التنفيذ من بداية النمط إلى نهايته .

٣ ــ عدم الانتقال إلى تعليمة داخل النمط، وإنما يكون الانتقال إلى بداية النمط أو نهايته حسب الحاجة .

٤ ــ عدم الانتقال من أية تعليمة داخل النمط المحدد إلى أية تعليمة أخرى خارجها، وإنما يكون الانتقال إلى نهاية النمط فقط .

خصائص وفوائد البرمجة التركيبية :

ينتج عن استخدام أساليب البرمجة التركيبية برامج ذات خصائص معينة، وفوائد بنيت عليها، يمكن تلخيصها كالتالى :

الخصائص :

١ ــ سهولة التحقق من صحة المعلومات المخرجة من البرنامج .

٢ ــ الوضوح وسهولة قراءة ومتابعة خطوات البرنامج، و بشكل خاص من قبل الآخرين .

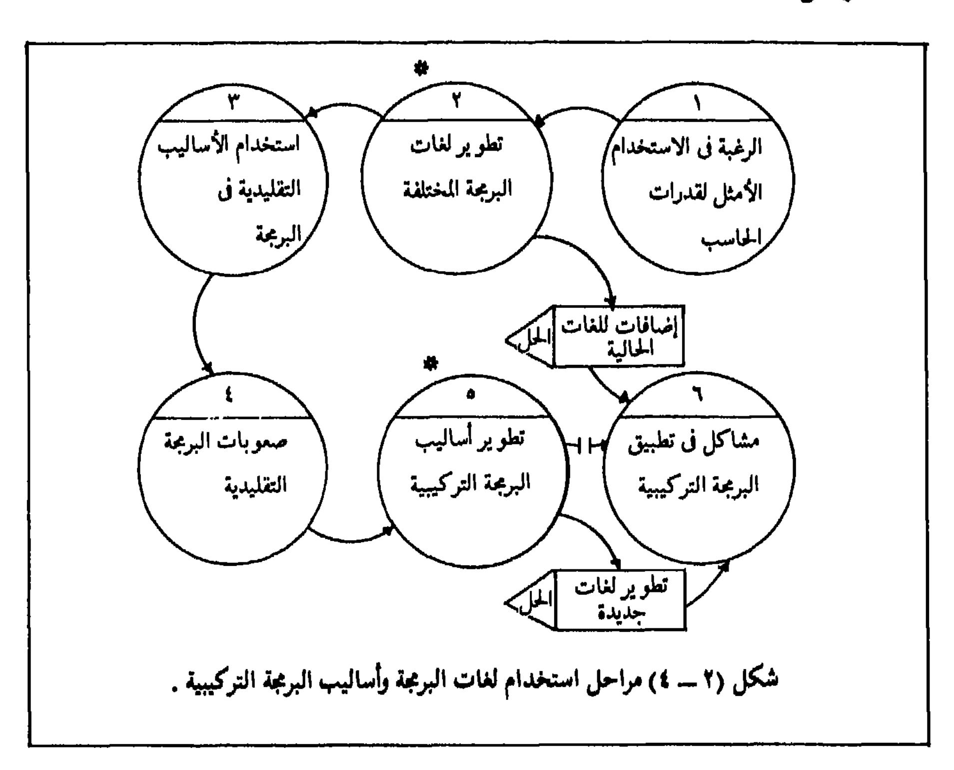
الفوائد :

١ ــ سهولة إجراء التعديلات على البرنامج .

٢ ــ زيادة إنتاجية المبرمجين بنسبة ١,٥ إلى ٢ .

المشاكل التى تعترض تطبيق البرمجة التركيبية :

نجد أن المشاكل التى اعترضت تطبيق البرمجة الهيكلية، جاءت نتيجة لمراحل التسلسل التاريخى التى مرت بها عملية التخاطب مع الحاسب، و يعكس الرسم التالى هذه المراحل :



شكل (٢ ــ ٤) مراحل استخدام لغات البرمجة وأساليب البرمجة التركيبية .

يلاحظ أن السبب الرئيسى فى ظهور المشاكل هوعدم التوافق بين تعليمات لغات البرمجة وأساليب البرمجة التركيبية، وذلك لكون اللغات قد طورت قبل الأساليب، كما هو مبين فى تسلسل الدوائر المؤشر عليها بــ « * » .

لذلك كانت حلقة الوصل المفقودة بين تطو ير الإطار النظرى لأساليب البرمجة التركيبية، وتطبيقها فى كل الظروف التى صاحبت تطو يرها، كما هو مبين فى شكل (٢ ــ ٤) الخط المقطوع بين الدائرتين رقم ٥ و ٦ .

وقد أخذت الجهود الرامية إلى معالجة مشاكل تطبيق البرمجة التركيبية (ب ت) اتجاهين :

١ ــ تعديل اللغات الحالية عن طريق إضافة تعليمات وخصائص تسهل تطبيق الـ (ب ت) .

٢ ــ تطوير برامج جديدة ذات تعليمات وخصائص توافق أسس وقواعد البرمجة التركيبية .

وقد ركزت الجهود على الحل الأول ؛ وذلك لسببين :

(١) للوقت الكبير المستثمر فى البرامج الحالية التى طورت باللغات الموجودة ،

(٢) لتوفر المبرمجين ذوى الخبرة فى استخدام هذه اللغات .

وقد واجه تنفيذ هذين الحلين الصعوبات التالية :

١ ــ صعوبة تغيير ما اعتاد عليه المبرمجون من أساليب البرمجة التقليدية مع توفر الرغبة فى التغيير .

٢ ــ صعوبة التغلب على خاصية الطبيعة البشرية فى مقاومة كل جديد ، سواء فى استخدام أساليب البرمجة التركيبية أو فى استخدام اللغات الجديدة .

٣ ــ صعوبة التزامن بين الوقتين اللازمين لاكتساب الخبرة فى اللغات الجديدة ، ولتلبية متطلبات المستفيدين ؛ إذ يتخلف الأول عن الثانى فى معظم الحالات .

٤ ــ صعوبة التوفيق بين الأساليب التقليدية فى البرمجة وأساليب البرمجة التركيبيةعند اجتماعهما فى برنامج واحد . و ينتج هذا عند اتباع أساليب البرمجة التركيبية فى التعديلات التى تتم على البرامج الحالية . وقد يؤدى هذا إلى إعادة كتابتها وفق أساليب البرمجة التركيبية .

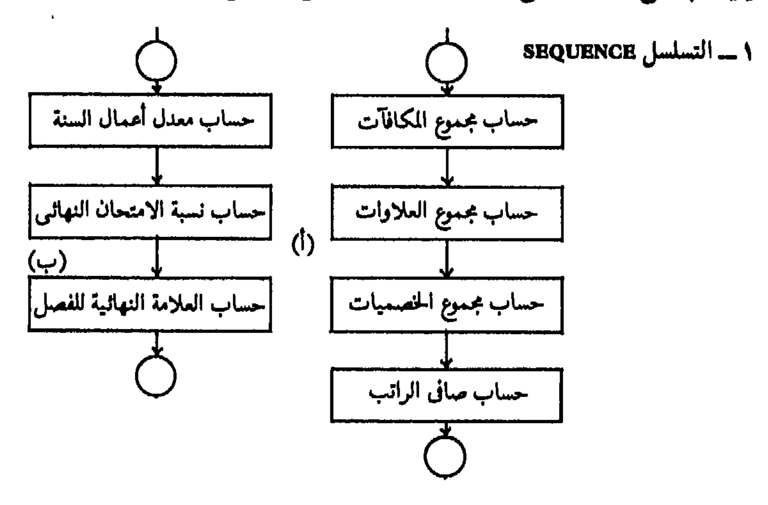
للتعرف على مدى قابلية أى من لغات البرمجة ، لابد من تحديد بعض الخصائص التى يجب أن تتصف بها :

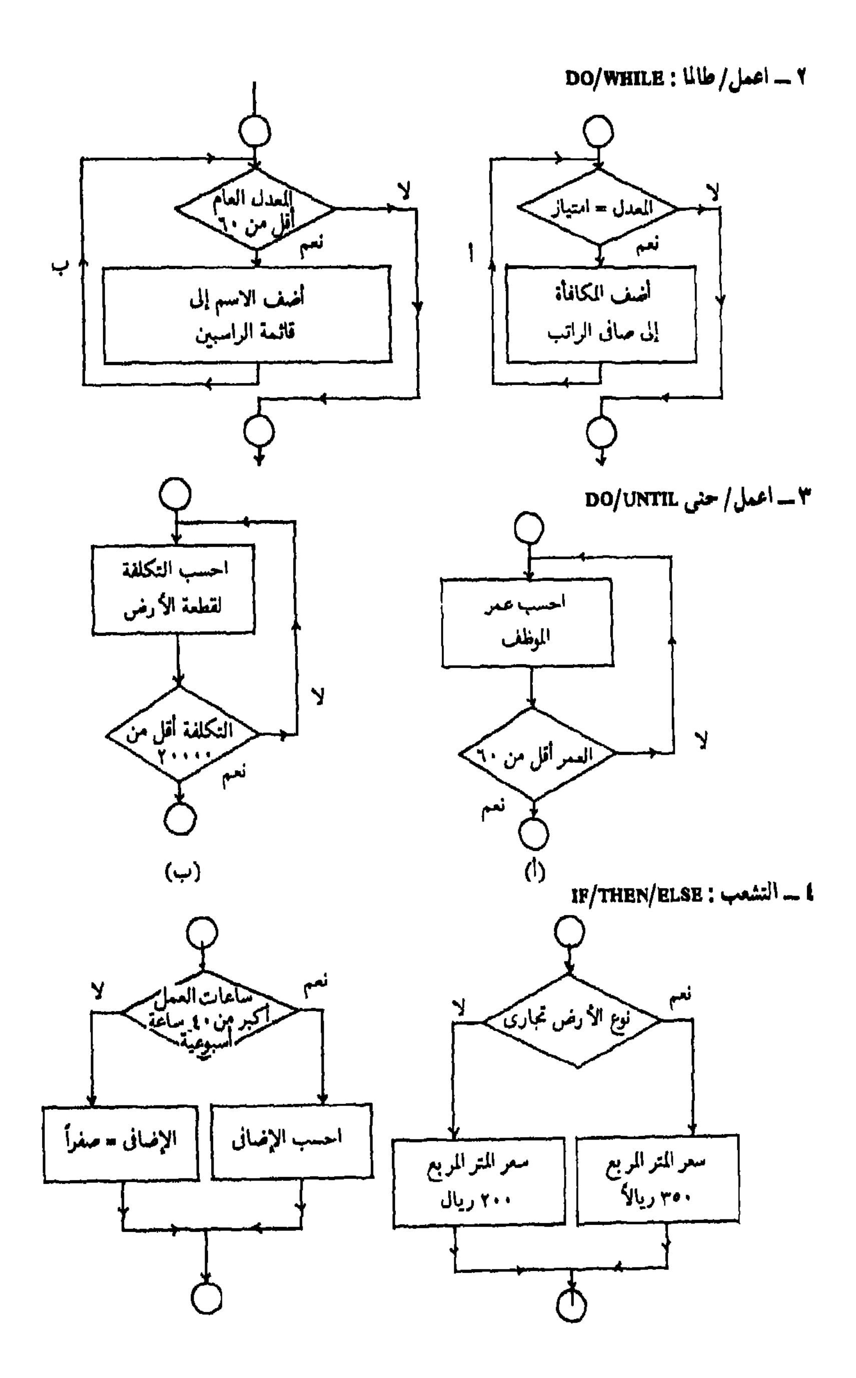
١ ــ تحديد نقطتى البداية والنهاية للنمط .
٢ ــ تنفيذ مجموعة من التعليمات المترابطة بتعليمة واحدة .
٣ ــ اختلاف مواقع بدء كتابة التعليمات فى الأسطر المترابطة ، وأن التحكم فى مساحة الهوامش بهذا الأسلوب يؤدى إلى زيادة الوضوح فى البرنامج .
٤ ــ تحديد بداية ونهاية تعليمات المقارنة المنطقية ، وخاصة المعقدة منها والتى تتطلب تسلسلاً متداخلاً للعديد منها ، و بطريقة منطقية .

ومن أمثلة اللغات التى قام الباحثون بتعديلات وإضافات عليها لتتكيف مع أساليب البرمجة التركيبية : الكوبول ، والفورتران ، والبيسك (لغة هذا الكتاب) ، فى حين تم تطوير لغة الباسكال ، مثلا كلغة متكاملة وفق أساليب البرمجة التركيبية . وقد لا توجد اللغات المعدلة أو الحديثة فى كل المؤسسات وعلى كل الحاسبات ، لذلك على المبرمج أن يحاول قدر استطاعته التقيد بأساليب البرمجة التركيبية ضمن اللغات المتاحة ، وذلك لتحقيق أكبر قدر ممكن من الاستفادة .

أمثلة مبسطة عن أساليب البرمجة التركيبية :

إليك بعض الأمثلة عن الأنماط المختلفة للبرمجة التركيبية :





٢ ـ اعمل/ طالما : DO/WHILE
المعدل = امتياز
لا
نعم
أضف المكافأة
إلى صافى الراتب
أ
المعدل العام
أقل من ٦٠
لا
نعم
أضف الاسم إلى
قائمة الراسبين
ب
٣ ـ اعمل/ حتى DO/UNTIL
احسب عمر
الموظف
العمر أقل من ٦٠
لا
نعم
(أ)
احسب التكلفة
لقطعة الأرض
التكلفة أقل من
٢٠٠٠٠
لا
نعم
(ب)
٤ ـ التشعب : IF/THEN/ELSE
نوع الأرض تجارى
نعم
لا
سعر المتر المربع
٣٥٠ ريالاً
سعر المتر المربع
٢٠٠ ريال
ساعات العمل
أكبر من ٤٠ ساعة
أسبوعية
نعم
لا
احسب الإضافى
الإضافى = صفراً

تمارين

١ ـ اختر النمط المناسب للمسائل التالية، ومن ثم ارسم الشكل الخاص به .

أ ـ تاجر يبيع نوعين من السجاد : العادى والفاخر .
سعر السجاد العادى ٢٠ ريالاً للقدم المربع، وللسجاد الفاخر ٣٥ ريالاً للقدم المربع .

ب ـ متعهد بناء يستمر بحساب مساحة مجموعة من البناءات إلا إذا زادت هذه المساحة على ٥٠٠ متر مربع .

جـ ـ يتم حساب مكافأة لكل عامل زادت كمية إنتاجه على ٣٠٠ قطعة .

د ـ للتوصل إلى تكلفة صناعة لتر واحد من «البوظة» تم حساب التكلفات التالية : الحليب، السكر، الزبدة، النكهة .

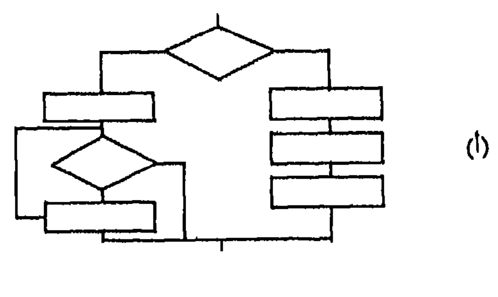
هـ ـ يعطى الزبون خصم ١٠٪ إذا زادت مشترياته الشهرية على ٥٠٠ ريال .

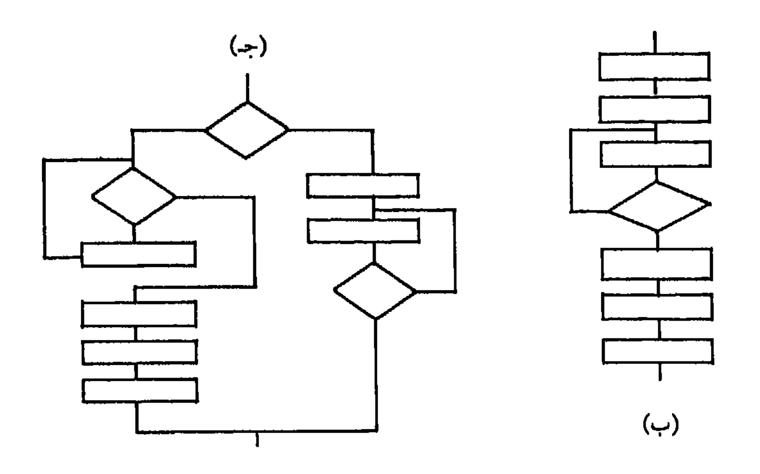
و ـ إذا كانت شهادة الطالب هى الثانوية، يضاف اسمه إلى قائمة الطلبة المؤهلين للالتحاق ببرنامج الدبلوم العادى، وإذا كان من حملة شهادة البكالوريوس يضاف اسمه إلى قائمة برنامج الدبلوم العالى .

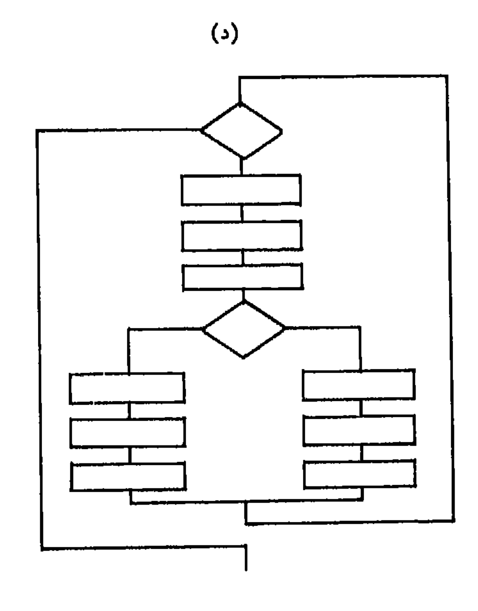
ز ـ يتم التوصل إلى صافى الربح بعد معرفة مجموع الإيرادات ومجموع المصروفات .

٢ ـ حدد نوع النمط من أساليب البرمجة التركيبية فى الأشكال المركبة التالية، وأضف الدوائر الصغيرة عند بداية ونهاية كل نمط :

(أ)




(جـ)
(ب)


(د)

الفصل الثالث

مقدمة عن التجزئة الوظيفية الهيكلية (الهرمية)

(Top-Down Functional Decomposition)

تعتبر التجزئة الوظيفية الهرمية الأساس الثانى الذى تستند إليه الأساليب الحديثة فى البرمجة، وذلك بالإضافة إلى البرمجة التركيبية التى تم شرحها فى الفصل السابق . ولعلنا نفرق هنا بين أسلوب البرمجة الذى يقضى بتقسيم البرنامج إلى أجزاء دون أن يكون هناك أى تخطيط سابق يرمى إلى ربط هذه الأجزاء بطريقة منطقية قبل تنفيذ عملية البرمجة الفعلية لهذه الأجزاء، و بين الأسلوب الذى يقضى بالدراسة المسبقة لمتطلبات المستفيدين من أجل الوصول إلى تعريف للوظائف الأساسية والفرعية، ومن ثم يتم تحديد الأجزاء التى تحقق هذه الوظائف وكيفية ترابطها بعضها مع بعض وفق قواعد متكاملة . هذه القواعد تتبع نمطاً معيناً يأخذ شكل الشجرة المعكوسة : أصلها فى السماء وفروعها متدنية إلى أسفل، أو ما يطلق عليه «التركيب الهرمى» .

من الممكن استخدام أسس البرمجة التركيبية دون أن يصاحب ذلك اتباع لأساليب التركيب الهرمى، مع أن البرمجة التركيبية بطبيعتها قد تفرض مستوى معيناً محدوداً من التجزئة البرمجية، و يتم ذلك عن طريق تحديد بدايات ونهايات كل جزء يتبع نمطاً معيناً من أنماط البرمجة التركيبية . والعكس صحيح أيضا، فقد تتبع أساليب التجزئة الوظيفية الهرمية دون أن تتبع معها أساليب البرمجة التركيبية . وأكثر ماينطبق ذلك على وضع تستخدم فيه إحدى لغات البرمجة التى لا تدعم أساليب البرمجة التركيبية بدرجة كبيرة . ولعل الوضع الأمثل هو فى أن تتكامل أساليب البرمجة الحديثة فيما بينها، حتى يتم تطوير البرامج وفق أسس كلا الأسلوبين، ومن ثم تتحقق الاستفادة المرجوة من استخدامها .

أسس التجزئة الوظيفية الهرمية

إن للتجزئة الوظيفية الهرمية أساسين رئيسيين : معرفة الوظائف المزمع القيام بها ، وترتيب هذه الوظائف بواسطة هيكل هرمى .

الأساس الأول : معرفة الوظائف :

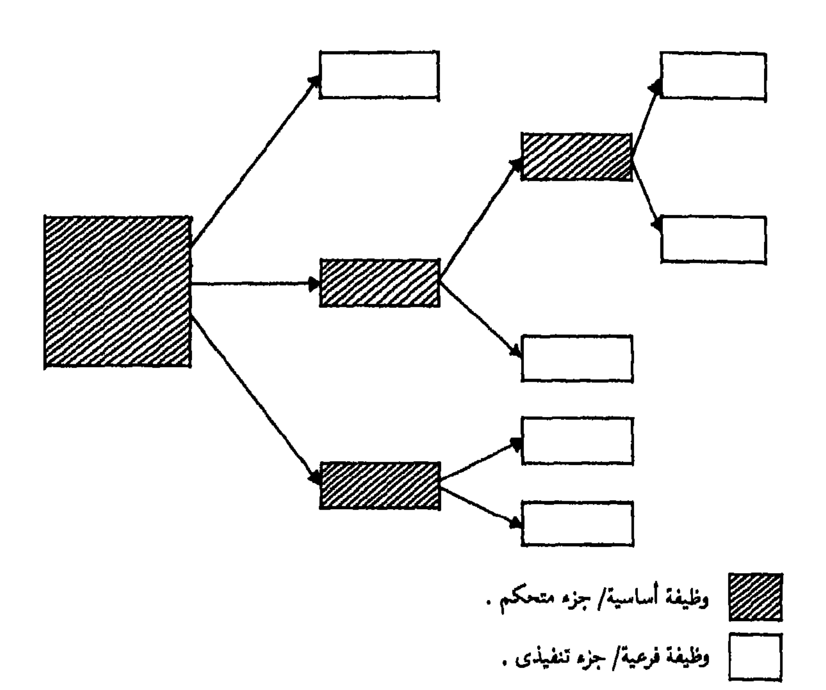
يبدو للوهلة الأولى أن الوصول إلى تحديد للوظائف أمر سهل . قد يكون هذا صحيحاً لدرجة ما إذا كانت متطلبات المستفيدين بسيطة وواضحة ولكن فى معظم الأحيان، لايكون ذلك واقعياً ، إذ كثيراً ما تكون الوظائف كثيرة ومتشعبة وغير واضحة المعالم . ويحتوى هذا الكتاب على أمثلة وتمارين ذات مستويات متعددة، ولما كان الهدف الرئيسى لها تعليمياً ، فإن الممارسة الفعلية تستدعى ممارسة وفهماً أعمق وأشمل مما تتطلبه هذه التمارين . وذلك مرجعه إلى طبيعة الملابسات والظروف التى تصاحب تطوير البرامج لحل مشاكل واقعية .

وسيحاول الكتاب عن طريق الحالات العملية إعطاء المتعلم صورة هى أقرب ماتكون إلى الواقع . وبيت القصيد هنا أن لايكوّن القارىء انطباعاً معيناً بعد الاطلاع على الأمثلة والتمارين، خاصة فى الفصول الأولى، بعدم جدوى اتباع هذا الأسلوب وبالتالى التقليل من أهميته . وستتم مناقشة بعض النقاط التى لها علاقة بهذا فيما بعد .

طبيعة دور الوظائف والأجزاء الخاصة بها :

من الوظائف ماتكون أساسية ومنها ماتكون فرعية . وتحقيق الهدف من الوظيفة الأساسية يسبقه تحقيق الأهداف المتعددة للوظائف الفرعية ويخصص لكل وظيفة جزء محدد .

وطبيعة الوظيفة تحتم الدور الذى يلعبه الجزء الخاص بتلك الوظيفة ، فالوظيفة الأساسية لها جزء التحكم Control Module والوظيفة الفرعية لها جزء التنفيذ Execution Module وتتفاوت مستويات التفرعات للوظائف حسب المتطلبات .
و يوضح الرسم التجريدى التالى هذه النقطة .



وقد يكون من الضرورى ، وإن كان من غير المستحب ، أن يقوم الجزء بوظيفة تنفيذية و وظيفة تحكم .
فقد يقوم الجزء بمعالجة تتطلب تفرعاً إلى جزء فرعى آخر أو أكثر ، ومن ثم الاستمرار فى المعالجة .
ومن الأفضل على كل حال أن يتلافى هذا الوضع عن طريق إعادة تركيب التفرعات ، وإضافة مستوى آخر من هذه التفرعات .

متى تتوقف التجزئة :

إن التحكم فى مستويات الأجزاء والوظائف التابعة لها أمر يخضع لاعتبارات شخصية، وليس لقوانين محددة تنطبق تلقائياً وفى كل الحالات .

وبالرغم من ذلك فإن هناك إرشادات عامة يمكن اتباعها للوصول بالتجزئة الوظيفية إلى مستوياتها المناسبة، وذلك بالإجابة على السؤالين التاليين :

- هل يقوم الجزء بوظيفة محددة واضحة بحيث إن الاستمرار بتجزئتها قد يؤدى إلى تشويه طبيعة هذه الوظيفة؟
- هل من الممكن برمجة الوظيفة التى يقوم بها هذا الجزء بسهولة ودون أن يتجاوز عدد عبارات البرمجة حداً أعلى (٥٠ عبارة مثلا)؟ ويحدد هذا العدد مسؤولية مركز الحاسب الآلى ليتم اتباعه من قبل جميع ذوى الاختصاص فى تطوير البرامج .

ومن الأخطاء الشائعة التى يقع فيها بعض المتعلمين عند التفكير فى طبيعة وظائف الأجزاء : تقسيمها إلى أجزاء تقوم بعملية إدخال البيانات، وأجزاء تقوم بعملية المعالجات، وأخرى تقوم بعملية إخراج المعلومات . وقد مر ذكر ذلك فى الفصل الأول، حيث إن هذه هى الدورة الرئيسية التى تتبعها خطوات معالجة البيانات على الحاسب .
ولكن ذلك لا يعنى بالضرورة أن تكون الأجزاء قد حددت بناء على هذه الوظائف الأساسية .

فالجزء قد يقوم بواحدة أو أكثر من هذه الوظائف الثلاث : الإدخال، والمعالجة، والإخراج، وذلك حسب ماتقتضيه طبيعة الوظيفة الكلية لأى برنامج .

ويكون التركيز هنا على طبيعة المعالجة دون عمليات الإدخال والإخراج . وكما مر ذكره، فإن المعالجة تتكون فى أساسها من العمليات الحسابية والمنطقية . وتتفاوت هذه فى صعوبتها وفق طبيعة المعالجة : من العمليات الحسابية البسيطة إلى العمليات

الإحصائية التنبئية المعقدة . وقد تجمع كل عمليات الإدخال والإخراج فى جزأين منفصلين ، وذلك إذا تطلبت المركزية فى تنفيذها . وفى الأحوال الأخرى ، و بشكل عام ، فإن توزيع عمليات الإدخال والإخراج إلى الأماكن التى تتم فيها المعالجات التابعة لها ، يؤدى إلى الوضوح فى طبيعة الوظيفة .

طبيعة التفاعل فيما بين الأجزاء :

يتم التفاعل والا تصال بين الجزء التنفيذى والجزء المتحكم بواسطة نوعين رئيسيين من المعلومات : المؤشرات والبيانات .

المؤشرات : تأخذ المؤشرات إحدى قيم متعددة ، وترمز كل قيمة منها إلى طبيعة الوظيفة الفرعية والجزء الخاص بها والذى سيتم التفرع إليه . وقد تستخدم المؤشرات كأسلوب للتحكم فى إصدار أمر بدء التنفيذ من الجزء المتحكم ، وإشعار الانتهاء من التنفيذ من الجزء التنفيذى . وفى الحالة الأخيرة ، تكون القيم ثنائية ، إما «نعم» أو «لا» ، فتكون القيمة «لا» عند إصدار أمر التنفيذ ، وتتحول إلى «نعم» عند الانتهاء منه ، وقد تمثل القيمتان بأرقام ، «٠» و «١» مثلاً .

البيانات : تتراوح كمية البيانات المتبادلة بين الأجزاء من الحقل (الوحدة الأساسية) إلى السجل . ومن الممارسات غير السليمة أن يتم استعمال ملف معين من قبل أجزاء متعددة ودون تخصيصها ، بحيث يصعب ملاحقة ومتابعة التغيرات التى تطرأ على الملف .

وكذلك فإن انسياب المعلومات بين الأجزاء يكون فى اتجاهين ، من الجزء المتحكم إلى الجزء المنفذ و بالعكس . فالبيانات المرسلة من الجزء المتحكم إلى الجزء المنفذ ، تحول إلى بيانات مخرجة ، ومن ثم يعاد إرسالها إلى الجزء المتحكم . وإذا كان هناك ضرورة فى جعل هذه البيانات مشتركة بين أجزاء متعددة فى النظام ، فليس ثمة حاجة إلى نقلها من جزء إلى جزء ؛ إذ قد تجعل مشاعة فى النظام ككل .

وإذا كانت المعلومات المتراسلة بين جزأين هى المؤشرات، فإن الجزء المتحكم يجعل قيمة المؤشر بإحدى قيمتيه «لا» مثلاً ــ و بعد انتهاء الجزء المنفذ من جميع المعالجات الخاصة به، يقوم بتغيير قيمة المؤشر إلى القيمة الأخرى له ــ «نعم» مثلا ــ معلناً بذلك انتهاء التنفيذ والعودة إلى الجزء المتحكم .

و بشكل عام يفضل التقليل من كمية البيانات المشتركة والمتحركة بين الأجزاء . ولكن قد يكون من الصعو بة بمكان، كتابة البرامج الجزئية مستقلة تماما إلى درجة انعدام أى تفاعل فيما بينها .

وفى كل الأحوال لابد من تحديد التفاعل والتشارك بين الأجزاء وتوثيقها بطريقة ما، سواء فى البرنامج نفسه أو فى التوثيق الملحق به .

حجم الجزء :

كان حجم الجزء يقاس تقليدياً بمقدار حجم الذاكرة بالحروف والذى يمكن أن يحتله الجزء فى الذاكرة . أما حديثاً، ومع انخفاض تكلفة الذاكرة، فإن حجم الجزء يقاس بعدد العبارات البرمجية القابلة للتنفيذ، أى باستثناء الملاحظات والتوجيهات والإيضاحات . وهناك مقياس آخر أدق يستند إلى هذا الأخير، وهوعدد الأسطر فى البرنامج بعد انضغاطه بإزالة بعض الحروف والفراغات .

وقد أشير إلى أن الاستمرار فى التجزئة يتوقف إذا كانت الوظيفة قابلة للبرمجة فى حدود (٥٠) سطراً من عبارات البرمجة . و ينطبق هذا على الأجزاء الفرعية التفصيلية التى تظهر فى البرنامج . لكن قد يكون حجم الجزء أكبر بكثير من ذلك، استناداً إلى مدى شمولية النظام .

ففى الشركات الكبيرة تقاس الأجزاء بآلاف السطور من عبارات البرمجة المنضغطة .

والتفريق هنا هو بين الجزء التابع لبرنامج والجزء التابع لوظيفة رئيسية فى شركة كبيرة . فالجزء الخاص بحساب مجمل الراتب هوجزء برمجى، والجزء الخاص بإنتاج حسابات شؤون الموظفين هو جزء رئيسى نتج عن ضرورة تحديد العلاقات بين الأجزاء الرئيسية الكبيرة فى مؤسسة ما . وهذا الجزء الأخير يكون فى العادة من نتاج التحليل والتصميم، أما الجزء البرمجى فقد ينتج عن نفس المرحلة أو مرحلة البرمجة . ذلك أن المبرمج قد يرى ضرورة وجود أجزاء صغيرة تقوم بوظائف بسيطة لتسهيل عملية كتابة وتنفيذ البرامج .

الأساس الثانى : التركيب الهرمى (TOP-DOWN) :

إن الوظائف الرئيسية والفرعية والأجزاء الخاصة بها لابد أن تتبع نظاماً محدداً لتنظيم العلاقات والتفاعل فيما بينها . و يتم تحقيق ذلك بواسطة التركيب الهرمى الذى يشبه الشجرة المقلوبة، كما مر ذكره فى الفصل السابق . وستتم الآن مناقشة بعض النقاط التى لها علاقة بهذا الأساس .

طرق رسم الهيكل الهرمى :

يبين الشكل التالى رسماً تجريدياً لتركيب هرمى

رس

١ أ
٢ ب
٣ جـ

١٠٢ د
٢٠٢ هـ
٣٠٢ ر
١٠٣ ح
٢٠٣ ط

١٠٣٠٢ ق
٢٠٣٠٢ ل

شكل (٣ ـ ١)
مثال تجريدى لهيكل هرمى.

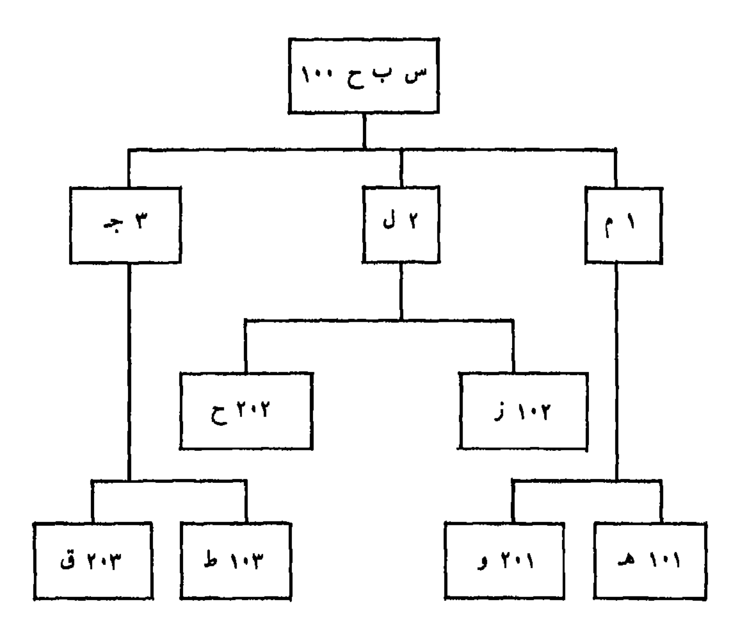
لاحظ النظام الذى اتبع فى ترقيم الأجزاء لتسهيل الإشارة إليها وربطها بعضها مع بعض . كذلك قد يعطى كل جزء اسماً مختصراً أو رمزاً معيناً مكوناً من أحرف و/ أو أرقام، حيث يتم استعمال هذا الاسم والرمز فيما بعد عند كتابة * البرامج . والأحرف المستخدمة فى أشكال التركيب الهرمى هى رمزية، ويمكن أن تكون فى أى تسلسل، ومكونة فى أغلب الأحيان من أحرف متعددة تشير إلى طبيعة الوظيفة .

أما فيما له علاقة بعدد المستويات، فإن التعداد الزائد لها فى صفحة واحدة قد يؤدى إلى تعقيدها و بالتالى صعوبة فهمها ومتابعتها، ولا يوجد عدد محدد متفق عليه للمستويات فى الصفحة الواحدة . وكإرشاد عام يمكن أن يكون الرقم من ٣ ـــ ٧ مستويات عدا المستوى الرئيسى الأول الذى يحوى مربعاً واحداً . ويمكن أن يؤخذ بعين الاعتبار التشعب الأفقى، إضافة إلى التشعب العمودى . و بشكل عام، فإن الزيادة فى التشعب الأفقى تؤدى إلى التقليل من عدد المستويات (التشعب العمودى) للصفحة الواحدة . وتعطى الصفحات التالية أمثلة تجريدية بهياكل هرمية . لاحظ التسلسل فى أرقام الأجزاء فى كل صفحة، والعلاقة فيما بينها على الصفحات المتعددة.

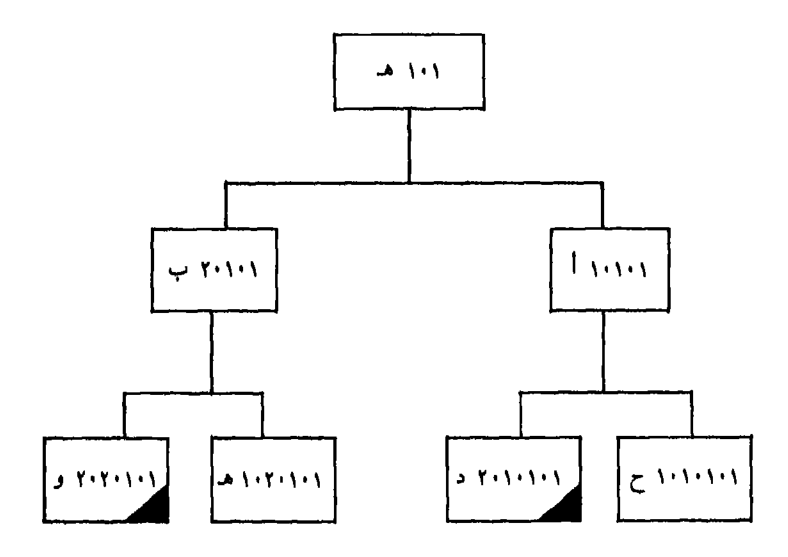
ومن فوائد هذا التقسيم أن ذوى العلاقة يطلعون على المستوى التفصيلى للتركيب الهرمى الملائم لاحتياجاتهم . فالمستفيد والمحلل والمصمم قد ينصب اهتمامهم على المستويات الشمولية، فى حين ينصب اهتمام المبرمج على المستوى التفصيلى للتركيب الهرمى، وخاصة عند صيانة البرنامج فى مرحلة التشغيل .

---

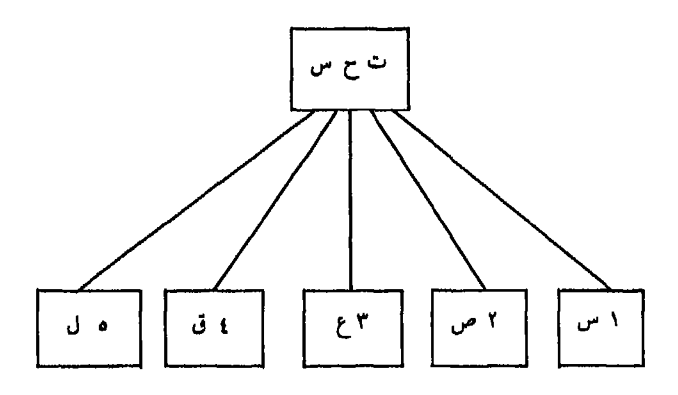
* عند كتابة البرامج بلغات البرمجة التى تستخدم عبارات باللغة الإنجليزية لمعالجة معلومات بالعربية، فإن هذه الأسماء والرموز تكون باللغة الإنجليزية .



شكل (٣ ـ ٢) المستوى الشمولي الأول في التركيب الهرمي .

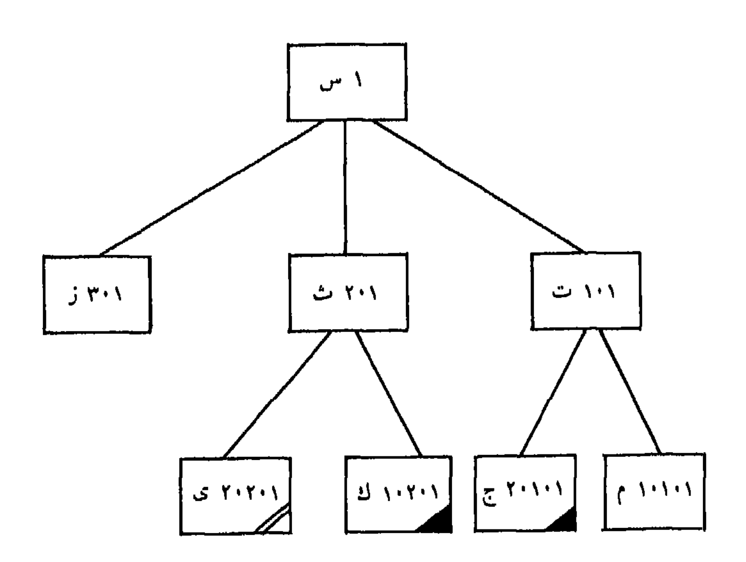


شكل (٣ ـ ٣) التركيب التفصيلي الهرمي للجزء رقم ١٠١ هـ من الشكل (٣ ـ ٢) .
و يستمر في تفصيل بقية الأجزاء من الشكل (٣ ـ ٢) حسبما تقتضيه طبيعة المشكلة .



شكل (٣ ــ ٤) تركيب هرمى ذو تفرعات أفقية متعددة

لاحظ أنه إذا دعت الحاجة إلى مستوى آخر لواحد أو اثنين من الأجزاء يمكن دمجها هنا. كذلك لاحظ الأسلوب الآخر لوصل الأجزاء.



شكل (٣ ــ ٥) تفصيل الجزء ١ س التابع لشكل (٣ ــ ٤)

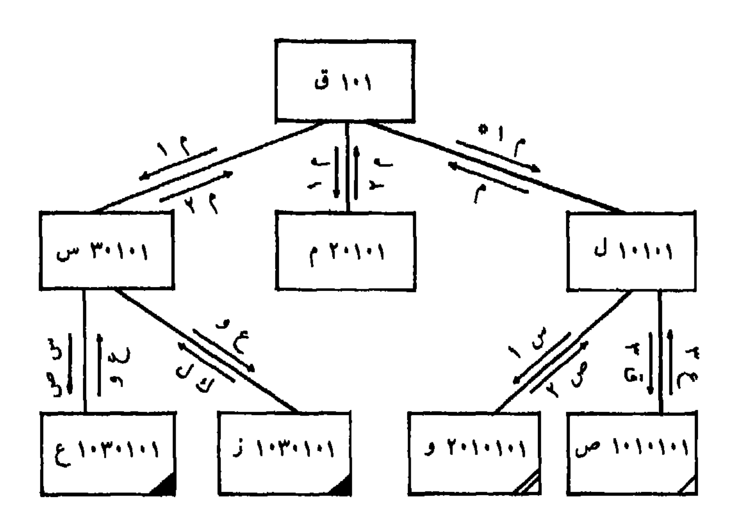
وهكذا لجميع الأجزاء الأخرى المبينة فى الشكل (٣ ــ ٤) إذا اقتضى الأمر .

ولتسهيل المتابعة فلا بأس من إضافة شروحات مبسطة لطبيعة الوظائف التى يقوم بها الجزء والممثلة بالرموز الموضوعة داخل المربعات وذلك فى أسفل كل صفحة، و/ أو قد توضع فى قاموس موحد خاص بالنظام ككل . (Data Dictionary) .

الأجزاء التى يتكرر تنفيذها :

وإذا تكرر تنفيذ جزء من قبل أكثر من جزء متحكم ، فهناك أساليب مقترحة لتبيان هذا التكرار : (أ) توضع إشارة موحدة للأجزاء المتكررة ذات الوظائف المختلفة فى إحدى الزاويتين السفليين، كما فى الشكل (٣ ــ ٣)، أى أن الجزء ٢٠١٠١٠١، والجزء ٢٠٢٠١٠١ ينفذان من قبل أجزاء متحكمة ، وليس من قبل جزء واحد فقط بغض النظر عن طبيعة الوظيفة للجزء المتكرر، (ب) تستخدم إما خطوط متعددة أو أحجام مختلفة من التظليل لتوحيد الإشارة إلى الأجزاء المتكررة ذات الوظيفة الواحدة، وفى نفس المجموعة من التركيبات الهرمية . انظر إلى الأمثلة فى الشكل (٣ ــ ٥) . أى أنه إذا تمت الإشارة إلى الوظيفة التى يقوم بها الجزء ٢٠٢٠١ فى تركيب هرمى آخر، فإن عدد الشركات يكون اثنتين، كذلك فإن حجم الجزء المظلل من المربع الممثل للجزء ٢٠١٠١ يماثل المربع الممثل لأى جزء آخر يؤدى نفس الوظيفة، و (جـ) الأسلوب الأخير هو المزج بين التظليل والخطوط لنفس المجموعة من التركيبات الهرمية . انظر إلى الأمثلة فى الشكل (٣ ــ ٦) . وقد يستخدم المصمم أو المبرمج نظماً أخرى شبيهة بهذه .

كاستخدام عدد نقاط مختلفة أو رموز مختلفة (ــ * ــ ◇) داخل المثلث فى زاوية المربع .



شكل (٣ ــ ٦)

استخدام مزيج من الأساليب للإشارة إلى الأجزاء المتكررة
التى تقوم بنفس الوظيفة فى تركيب هرمى تفصيلى .

تحتوى المربعات الأخرى الممثلة للوظيفة ز نفس الحجم المظلل .

* هناك إشارات ورموز أخرى سيتم التحدث عنها تباعاً .

ويمكن الاكتفاء باستخدام نفس الرمز للوظيفة الواحدة المتكررة للحفاظ على وحدة الإشارة إليها فى التركيبات الهرمية المختلفة التابعة لنظام واحد .

البيانات المشتركة بين الأجزاء :

ومن الاتجاهات الحديثة فى رسم التركيب الهرمى، تبيان البيانات على الخطوط الموصلة بين الأجزاء . وهناك تعبير جديد يطلق على التركيب الهرمى الموضح عليه انسياب البيانات وهو «الخريطة التركيبية» (Structure Chart) .

وكمثال على ذلك، انظر إلى الشكل (٣ ــ ٦) ولاحظ الإشارات والكتابة فوق الخطوط الموصلة بين الأجزاء . وتدل الرموز المبينة فوق الخط م ١ على اسم البيانات

المنقولة من وإلى الجزء المتحكم . أما النقطة السوداء فى نهاية الخط فتشير إلى اتجاه انسياب البيانات . ومن الأفضل تبيان البيانات التى تتم معالجتها فقط فى الجزء المنفذ .

فمثلا ، إذا كانت البيانات المنقولة سجلاً كاملاً للموظف ، ولكن الجزء المنفذ يستخدم حقل الدرجة والمرتبة فقط ، يجب ذكر الحقلين الأخيرين فقط .

و يؤدى اتباع هذا الأسلوب إلى الوضوح فى الخريطة التركيبية وسهولة متابعة وظائف أجزائها .

التسلسل فى تنفيذ الأجزاء :

تخضع الأجزاء حين تنفيذها إلى أحد أسلوبين : الاختيارى ، والمتتالى . فأما الاختيارى فيقضى بأن الجزء المتحكم يختار أحد الأجزاء المنفذة دون الآخر . وإذا كان الأمر كذلك ، يوضع رمز الاختيار فى أسفل منتصف المربع الممثل للجزء . انظر إلى الجزء ١٠١ ــ ق فى الشكل (٣ ــ ٦) وشرح ذلك ، أن الجزء «ق» ينفذ أحد الأجزاء الثلاثة «ل» أو «م» أو «س» بناء على قيمة المتغير م ١ ، وترسل قيمة محددة للبيانات م ٢ من الجزء المنفذ إلى المتحكم .

وأما فى أسلوب المتتالى ، فيتم تنفيذ الأجزاء المتشعبة من جزء متحكم بالتتالى . ففى الشكل (٣ ــ ٦) ينفذ الجزء «و» ومن ثم الجزء «ز» . وفى بعض الحالات قد يكون المخرج من جزء منفذ هو بذاته المدخل إلى جزء آخر . ففى نفس الشكل ، نجد أن البيانات «ع و» هى مخرجات من الجزء «ع» ومدخلات إلى الجزء «ز» .

بالإضافة إلى ذلك ، فإن الانتقال من جزء إلى آخر يجب أن يتم عبر السلم الهرمى للأجزاء . فللانتقال من الجزء «ز» إلى «و» ، يجب اتباع المسار التالى :

ز ←── س ←── ق ←── ل ←── و .

خصائص التجزئة والوظيفية الهرمية :

كما سبق ، يتبين أن من خصائص التجزئة الوظيفية الهرمية ما يلى :

١ ـ تساعد على تفتيت المشكلة المعقدة الكبيرة إلى وحدات صغيرة يسهل التحكم فيها .

٢ ـ تعطى صورة واضحة عن النظام قيد الدراسة ، سواء كانت شاملة أو تفصيلية ، وذلك حسب اهتمام المطلع عليها .

٣ ـ تبين الوظائف الأساسية والفرعية وكيفية ترابطها فيما بينها .

٤ ـ تعطى صورة عن الوضع الحالى للنظام ، وذلك لسهولة تحديثها وصيانتها ، سواء بالإضافة أو بالحذف أو بالتعديل .

فوائد التجزئة الوظيفية الهرمية :

لأسلوب التجزئة الوظيفية الهرمية فوائد متعددة :

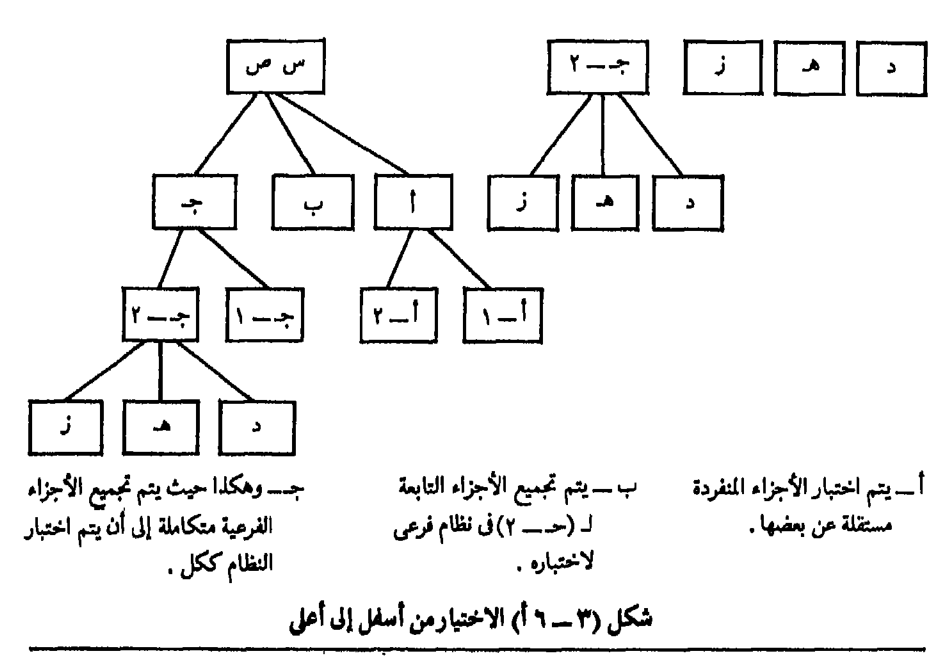
١ ـ تساعد فى عملية تطوير وكتابة البرامج . وتتم هذه العملية بإحدى خطتين : التنفيذ من أعلى إلى أسفل ، أو من أسفل إلى أعلى .

والتنفيذ من أعلى إلى أسفل يتطلب كتابة البرامج للأجزاء العلوية ، متفرعاً إلى المستويات التنفيذية السفلية ، وعند كل مستوى يتم استخدام برامج وهمية لأجزاء المستوى الذى تحته ، إلى أن يتم إنهاء جميع الأجزاء . ففى الشكل (٣ ـ ٤) مثلاً ، يتم تطوير البرنامج للأجزاء ت ح س ، س ، ص ، ع ، ق ، ل كمرحلة أولى . و بعد ذلك قد تتبع البرمجة أحد اتجاهين : إما أن يتم التطوير لجميع الأجزاء فى المستوى التالى ، مستوى ت ، ث ، ز فى الشكل (٣ ـ ٥) ، وهكذا التحرك أفقياً وإنهاء برمجة الأجزاء كلها فى مستوى معين . وإما أن يكون الاتجاه عمودياً ، كما فى الشكل (٣ ـ ٥) مثلاً ، حيث يتم الانتهاء من برمجة جميع أجزائه ، مستوى مستوى ، إلى آخر مستوى سفلى ، م ، ج ، ك ، ى ، فى هذه الحالة .

ومن جهة أخرى ، فإن التنفيذ من أسفل إلى أعلى يقضى بكتابة البرامج للأجزاء التنفيذية فى أدنى المستويات ، حيث تجمع هذه الأجزاء بعضها مع بعض وفق التركيب الهرمى ، إلى أن يتم تجميع جميع الأجزاء للنظام ككل .

٢ ــ تسهيل عملية اختبار النظام . وكما فى تطوير البرامج ، فإن اختبار النظام يتبع أحد الأسلوبين : من أعلى إلى أسفل ، أو من أسفل إلى أعلى .[1]

وربما يكون من المستحسن اختبار نفس الأسلوب لتطوير واختبار النتائج . وإن كان ذلك ليس ضرورياً . ففى حالة اتباع أسلوب التطوير من أعلى إلى أسفل و بشكل عمودى ، يمكن البدء بالاختبار به من أسفل إلى أعلى للجزء من التركيب الهرمى الذى تمت كتابة البرامج له .



أ ــ يتم اختبار الأجزاء المنفردة مستقلة عن بعضها .

ب ــ يتم تجميع الأجزاء التابعة لـ (جـ ــ ٢) فى نظام فرعى لاختباره .

جـ ــ وهكذا حيث يتم تجميع الأجزاء الفرعية متكاملة إلى أن يتم اختبار النظام ككل .

شكل (٣ ــ ٦ أ) الاختبار من أسفل إلى أعلى

1. Edward Yourdan, ((Techniques of Program Structure and Design)) (Englewood Cliffs : Printic Hall , Inc., 1975), p.58.

٣ ــ تسهيل عملية اكتشاف الأخطاء وتحديد آثارها . إن تقسيم المشروع الكبير المعقد إلى أجزاء وظيفية ، وتقليل التفاعل فيما بينها ، وتعريفه ، يؤدى إلى سهولة اكتشاف موطن الأخطاء وتحديد مسار تأثير خطأ مافى جزء وظيفى على الأجزاء الأخرى .

وكلما زاد مستوى التفاعل بين الأجزاء ، كلما زادت معه صعوبة اكتشاف وعزل وتصحيح الأخطاء ، وخاصة إذا مانتج هذا الخطأ عن تعديل فى جزء ما فى مرحلة التشغيل ، للنظام ككل . و يطلق على انتشار تأثير الخطأ فى جزء إلى الأجزاء الأخرى «العامل التموجى» .

بالإضافة إلى ذلك فإن تأثير التعديلات قد يسبب مشاكل أيضا ، فتعديل فى طبيعة المدخلات أو طبيعة المعالجات قد يقتضى تعديلات أخرى فى أجزاء لها علاقة بالجزء المعدل .

فتحديد وظائف الأجزاء ، وتقليل وتقنين التفاعلات فيما بين الأجزاء ، يسهل عملية عزل واكتشاف الأخطاء ، ويخفف من تأثير التعديلات فى جزء على الأجزاء الأخرى .

٤ ــ تساعد فى إنجاز عمليات الصيانة . يمكن تقدير أهمية هذه الفائدة إذا علم أن حوالى ٥٠٪ من الموارد المالية والبشرية المخصصة لتشغيل وتطوير البرامج تستهلك فى صيانة البرامج الحالية[٢] . فالأجزاء الجديدة تضاف إلى الهيكل الهرمى فى موقع مناسب ، حيث تتوافق وتتجانس وظيفة الجزء الجديد مع وظائف الأجزاء الأخرى . أما طلبات الصيانة التى تستدعى تعديل جزء خالٍ ، فيتم توجيهها إلى الجزء ذى العلاقة .

---

2 — B. P. Lebtz, E. B Swanson and, G. E. Tompkins, "Characteristics of Application Software Maintenance," Communication of the ACM. Vol. 21, No. 6 (June 1978), 466-471.

وكذلك نتائج استبيان وزع على حوالى ٢٣ مركزاً رئيسياً للحاسبات فى مدينة الرياض عام ١٤٠٥ هـ . انظر ندوة «التقنية الحديثة فى تنظيم وإدارة المعلومات» ، معهد الإدارة العامة ، ٥ ــ ٦ ربيع الآخر ١٤٠٧ هـ ، صفحة ٥٠ .

و يتم اختبار الجزء الجديد أو المعدل دون الحاجة إلى اختبار النظام ككل . وهذا كله يؤدى ، بالإضافة إلى التوفير فى المال والوقت ، إلى التخفيف من العبء النفسى على مبرمجى الصيانة .

٥ ـــ تستخدم كأداة لإدارة مشاريع التطوير، حيث يمكن استخدام الهيكل الهرمى كوسيلة للتحكم فى توزيع الصلاحيات ، ولتحسين وسائل الاتصال بين المجموعات المختلفة التى تقوم بتطوير نفس النظام ، ولتقويم نتائج المشروع ومدى تقدمه نحو تحقيق هدفه .

ففى المشاريع المعقدة الكبيرة ، يمكن توزيع مسؤوليات تطوير الأجزاء على مجموعات مختلفة وفق التركيب الهرمى . و يتم الاتصال بين هذه المجموعات ؛ لتحديد أى تفاعل بين الأجزاء المطورة . وفى أية لحظة من لحظات تطوير المشروع ، يمكن لمدير المشروع أن يحصل على فكرة عامة عن مدى تقدم الأجزاء ، وعما إذا كان ماتم إنجازه من الأجزاء قد حقق الهدف من وظيفتها .

٦ ـــ زيادة إنتاجية المبرمج بعامل مقداره ١,٦ [٣] .

---

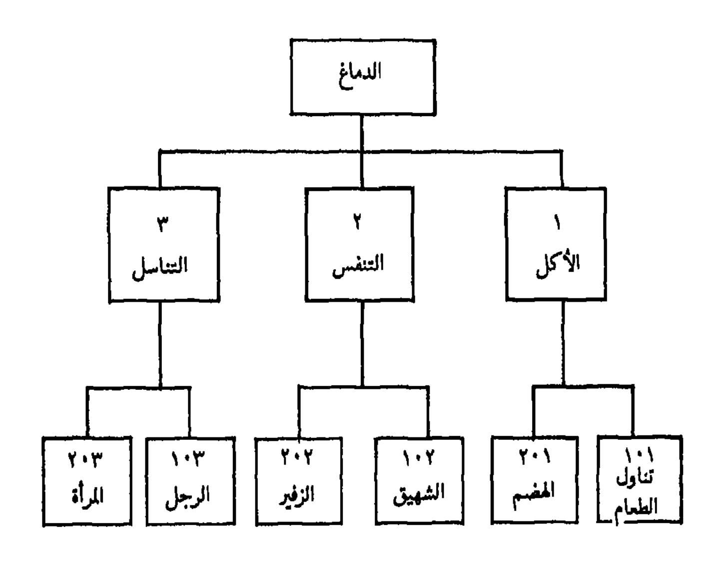
3 — E. Yourdon,« Managing the Stuctured Techniques,» New York, pres, 1984 , p. 33

أمثلة عملية

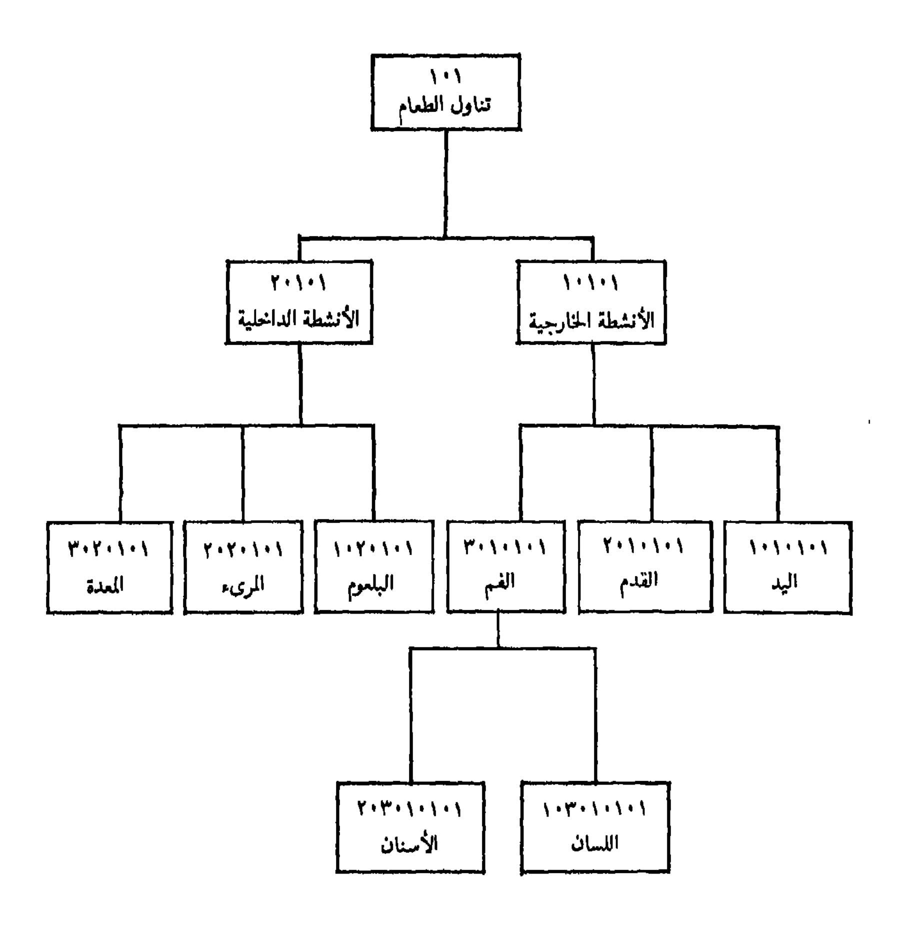
تقدم الصفحات التالية أشكالاً متعددة من التركيب الهرمى لأمثلة مختلفة .
وسيكون تسلسل الأمثلة كالتالى :
الإنسان، مثال حكومى، مثال تجارى .

الإنسان

يؤدى الإنسان وظائف متعددة ومتداخلة، وهناك وظائف أساسية كالتنفس والأكل والتناسل، وهناك وظائف فرعية تقوم بها أجزاء تابعة لكل وظيفة رئيسية كالأيدى والأنف واللسان . و يعطى الشكل (٣ ــ ٧) الهيكل الهرمى العام للإنسان، ويمثل الشكل (٣ ــ ٨) الهيكل الهرمى التفصيلى لوظيفة الطعام .



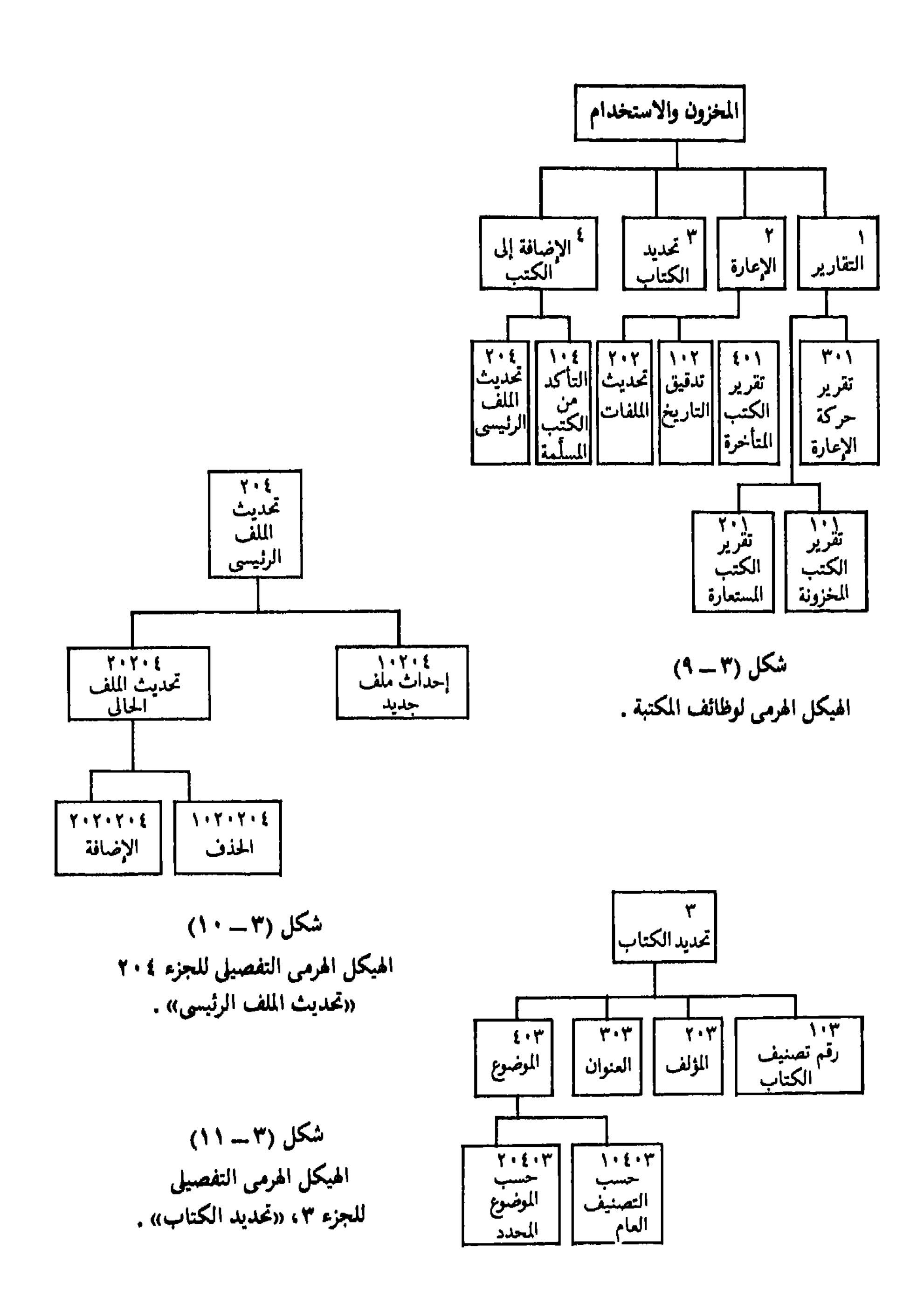
الشكل (٣ ــ ٧) الهيكل الهرمى العام لوظائف الإنسان



شكل (٣ ــ ٨) الهيكل الهرمى التفصيلى للجزء ١٠١ «تناول الطعام»

مثال حكومى :

تحتوى مكتبة مدرسة حكومية ثانوية على مايقارب ٢٠٫٠٠٠ كتاب ، وتتبع تصنيف ديوى . وقد قام فريق بتحليل عمليات وأنشطة المكتبة ، وتم تصميم عدة هياكل هرمية متسلسلة لوصف الوظائف الرئيسية فى النظام ، كما هو مبين فى الأشكال (٣ ــ ٩) ، (٣ ــ ١٠) ، (٣ ــ ١١) .



شكل (٣ ــ ٩)

الهيكل الهرمى لوظائف المكتبة .

شكل (٣ ــ ١٠)

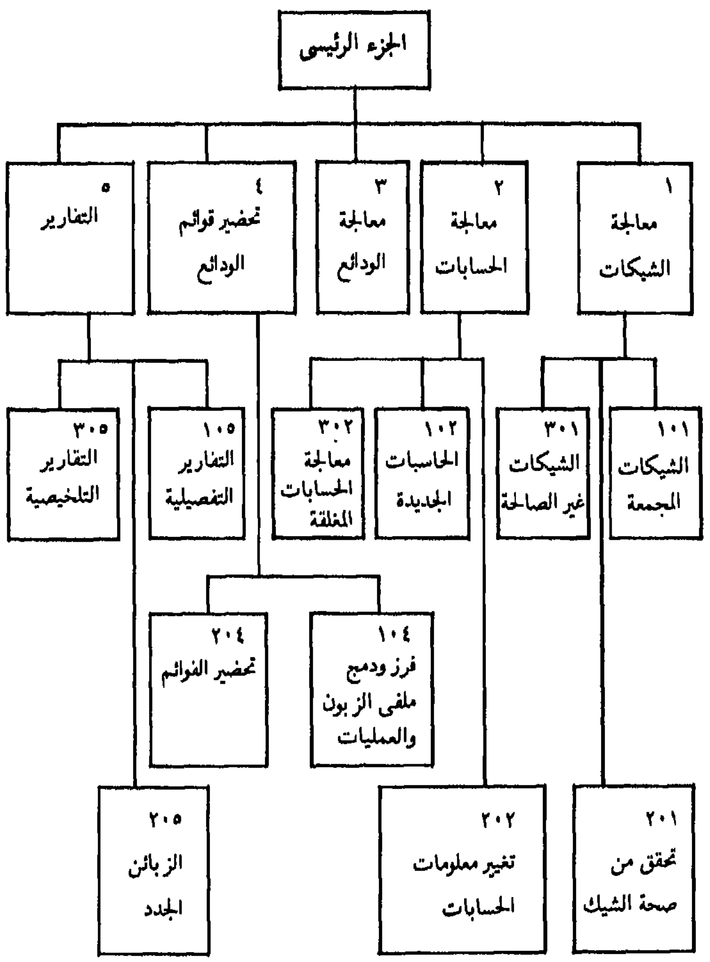
الهيكل الهرمى التفصيلى للجزء ٢٠٤ «تحديث الملف الرئيسى» .

شكل (٣ ــ ١١)

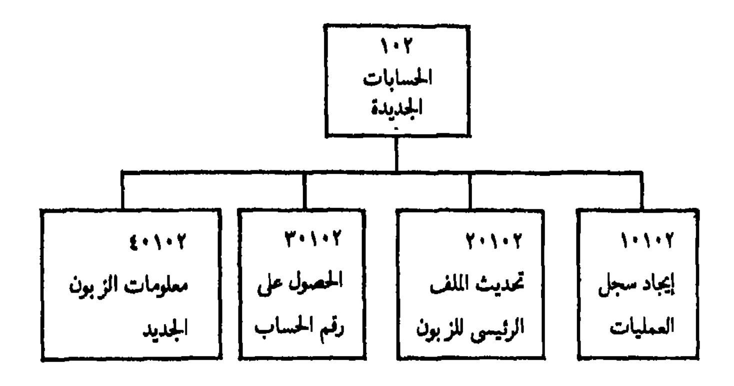
الهيكل الهرمى التفصيلى للجزء ٣، «تحديد الكتاب» .

مثال تجارى :

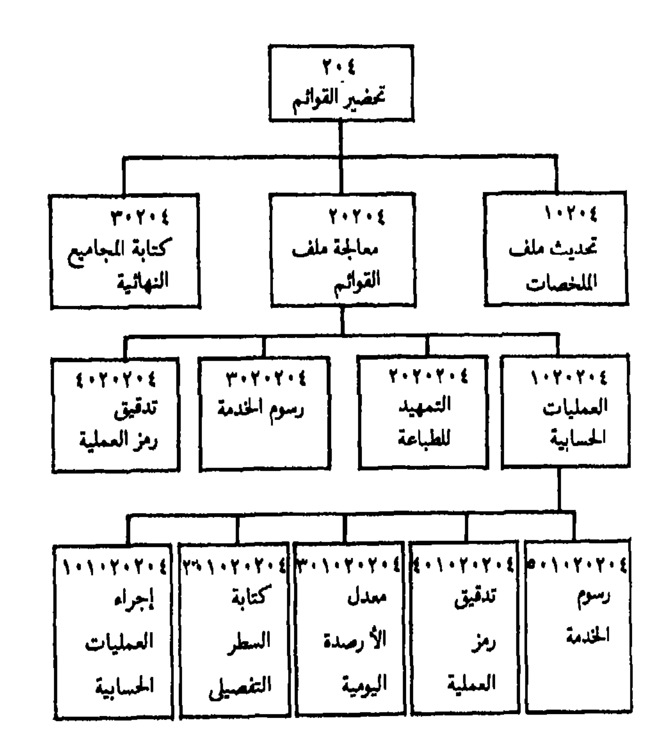
يقدم البنك س ص ع خدمات متعددة، وقد أرادت الإدارة إضافة خدمة جديدة لنوع معين من الحسابات التى تحمل فائدة، وتخضع لقواعد محددة . وقد تم تحديد عدة هياكل هرمية بوصف وظائف النظام وعلاقاتها بعضها مع بعض، ويحتوى الأشكال (٣ ـ ١٢)، (٣ ـ ١٣)، (٣ ـ ١٤) بعضاً منها .



شكل (٣ ـ ١٢) الهيكل الهرمى العام لمعالجة الحسابات الجارية الخاصة ببنك س ص ع.



شكل (٣ ـ ١٣) الهيكل الهرمى التفصيلى للجزء ١٠٢ «الحسابات الجديدة» .



شكل (٣ ـ ١٤) الهيكل الهرمى التفصيلى للجزء ٢٠٤ «تحضير القوائم» .

تمارين

١ ــ ماهى طبيعة تسلسل تنفيذ الأجزاء ؟

٢ ــ اشرح باختصار أنواع الأجزاء فى الهيكل الهرمى .

٣ ــ اشرح باختصار الأساسين الرئيسيين للتجزئة الهرمية .

٤ ــ ماهى أساليب تطوير (كتابة) البرامج للأجزاء المبينة فى الهيكل الهرمى؟

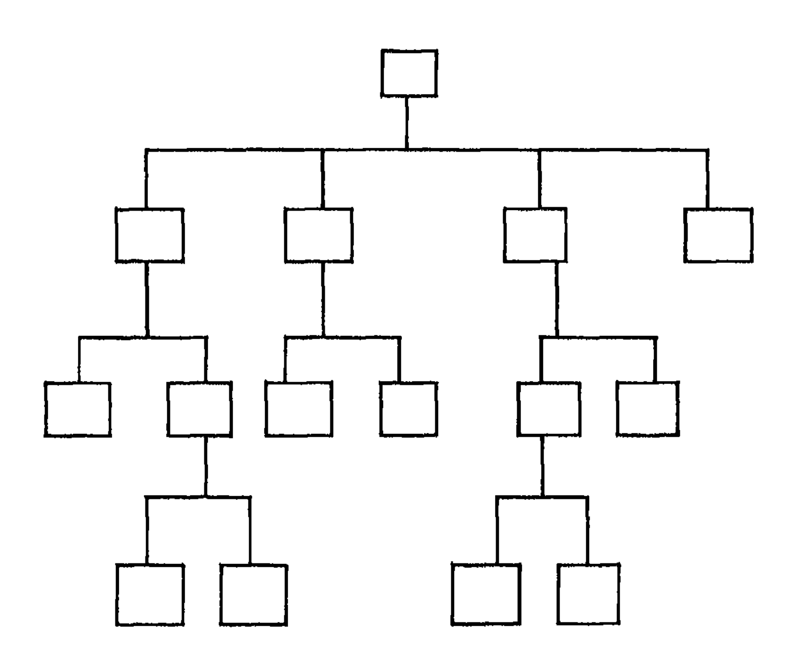
٥ ــ ماهى أساليب اختبار الأجزاء فى الهيكل الهرمى؟

٦ ــ للهيكل الهرمى التالى أجب عن الأسئلة الآتية :

أ ــ رقم الأجزاء ، وأعط أسماء لها مكونة من حرفين .

ب ــ حدد الأجزاء التنفيذية والأجزاء المتحكمة .

جـ ــ كيف يتم تطوير البرامج لهذه الأجزاء ؟ اذكر جميع الأساليب المتوفرة .



٧ ــ التعليمات : ارسم الهيكل الهرمى للحالات التالية حسب ماتراه مناسباً ، سواء فى مستوى واحد أو مستو يات متعددة . ضع فرضيات مقبولة لأى معلومات ناقصة .

أ ــ تشتمل عمليات الطلبيات فى محل (هارب للأسماك) على النشاطات الرئيسية التالية :

يقدم الز بون طلباً ، ومن ثم يرى إذا كان فى الإمكان منحه الدَّيْن المطلوب فى الطلبية . أما عمليات البضائع فتتكون من فاتورة الز بون ، أوامر الشحن للبضائع ، مذكرة رفض من الز بون ، وتحديث ملف المخزون . أما طلبية الز بون فتشمل رقم الحساب والأسماك المطلوبة ، و يتم إيجاد التكلفة وشروط البيع .

ب ــ يشتمل نشاط قسم خدمات الأغذية فى مستشفى كبير على عمليتين رئيسيتين : تحضير الطعام ، والتحكم فى عيارات الأكل عند المرضى ، والمرضى الذين لهم عادة (حمْية) معينة إما أن يكونوا على برنامج عام للأكل أو خاص . والبرنامج الخاص يتطلب حساب متطلبات التغذية والكالوريات والبروتينات .

جـ ــ تشتمل العمليات الرئيسية لمراقبة المخزونات فى مخازن تميم الكبيرة والمتعددة الفروع على التالى :

ــ التسعير .
ــ التسلم .
ــ الطلبات العامة .
ــ طلبات التنزيلات .

أما الطلبيات العامة فيمكن أن تكون فى أحد الأشكال التالية :

ــ طلبيات الملابس عن طريق التذاكر الخاصة بها .
ــ طلبيات البضائع الأخرى من الموردين .
ــ الطلبيات الأ توماتيكية عن طريق جهاز الفحص الإلكترونى .

الفصل الرابع

مقدمة عن وسائل تصنيف البيانات والمعلومات فى الحاسب

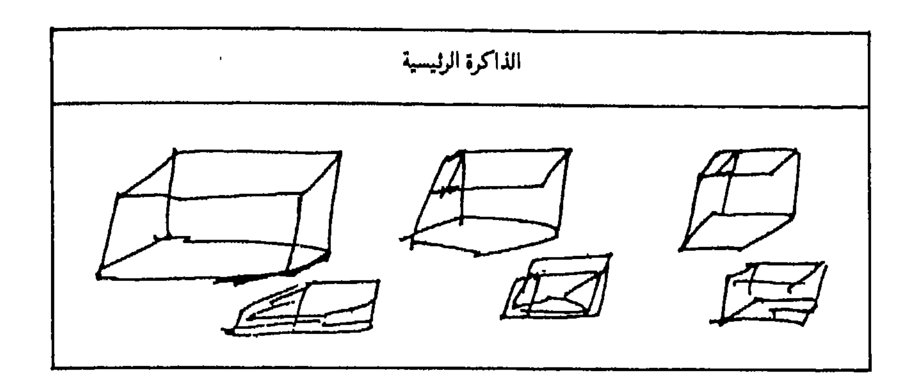
كما مر شرحه فى الفصل الأول، فإن البيانات تدخل إلى الحاسب عن طريق أحد أجهزة الإدخال وتخزن فى الذاكرة، ومن ثم تخضع للمعالجة الحسابية أو المنطقية أو كلتيهما، وتخرج النتيجة كمعلومات على إحدى وسائط الإخراج : شاشة، طرفية، أو طابعة مثلا .

و يصحب عملية إدخال البيانات إدخال للبرنامج الذى يحدد الخطوات التى ستخضع لها البيانات بإحدى لغات البرمجة .

وتتبع طرق متباينة فى صعوبتها لوصف العمليات التى يقوم بها الحاسب عند القيام بالوظائف الأساسية الثلاث : الإدخال، المعالجة، والإخراج . وتزداد هذه الصعوبة كلما اقترب الوصف من عمل الحاسب الفعلى، وتقل الصعوبة كلما اقترب هذا الوصف من أسلوب عمل المبرمج من خلال لغات البرمجة، وتمشياً مع الأسلوب العام المبسط لهذا الكتاب، فإن وصف هذه العمليات سيتم عن الطريق الأقل صعوبة وتعقيداً، وهو من وجهة نظر المبرمج .

وصف عام لأسلوب تعامل الحاسب مع البيانات :

ينظر المبرمج إلى ذاكرة الحاسب وكأنها مقسمة إلى صناديق تخزن فيها البيانات . هذه الصناديق تحوى البيانات عند معالجتها الفعلية من قبل الحاسب، أى عند تخزينها فى الذاكرة الرئيسية تمهيداً لمعالجتها . و يوضح الشكل التالى مفهوم الصناديق .



ومن الممكن القيام بالعمليات التالية للتحكم فى محتو يات هذه الصناديق :

١ ــ **إدخال وتخزين البيانات بداخلها**، و يتم ذلك إما عن طريق وسائط الإدخال والأ وامر التابعة لها، أوعن طريق التخزين المباشر فى داخل البرنامج والأ وامر التابعة له . والبيانات المدخلة عن طريق وسائط الإدخال إما أن تدخل عن طريق لوحة المفاتيح التابعة للطرفيات، أوعن طريق وحدات الذاكرة الفرعية المساعدة : الأشرطة، والأقراص الممغنطة . ففى الحالة الأ ولى يكون الإدخال فى وضع التخاطب مع الحاسب، وفى الحالة الثانية يتم تخزين البيانات على الأشرطة والأقراص الممغنطة أولاً، ثم تسترجع بعد ذلك لمعالجتها من قبل البرنامج .

و يتم التخزين الأ ولى هذا إما عن طريق أجهزة خاصة تخزن البيانات فى الذاكرة الفرعية مباشرة من لوحة المفاتيح، أوقد يكتب برنامج خاص لذلك الغرض .

٢ ــ **القيام بالعمليات الحسابية** : إذا احتوت هذه الصناديق على بيانات رقمية، يمكن إخضاعها للعمليات الحسابية المختلفة، من جمع وطرح وقسمة وضرب وأس .

ويمكن أن تؤلف محتو يات الصناديق قيماً فى معادلات معقدة وطو يلة، حيث يقوم الحاسب بتقسيم هذه العمليات إلى سلسلة من الحسابات المرحلية وفق أسس معروفة ستناقش لاحقاً فى هذا الفصل .

٣ ــ **نسخ وتبديل محتويات الصناديق من البيانات :** من الممكن نقل محتويات صندوق إلى آخر، أو تغيير محتويات الصندوق الواحد، أو تبادل القيم بين صندوقين .

٤ ــ **إخراج محتويات الصناديق إلى الأجزاء الملحقة بالمعالج :** سواء كان ذلك إلى الشاشة التابعة للطرفية، أو إلى آلة الطباعة، أو إلى وسائط التخزين المساعد .

كيفية التحكم فى البيانات المخزنة فى الصناديق :

يتسم هذا التحكم عن طريق إتاحة الفرصة للمبرمج بإعطاء أسماء لهذه الصناديق تتكون من أحرف وأرقام . وتختلف قواعد تسمية هذه الصناديق من لغة إلى أخرى، وإن كان الاتفاق بينها جميعاً أن تبدأ بحرف هجائى .

وعلى المبرمج أن يتابع محتويات هذه الصناديق وموافقتها لأسمائها إذ أن الحاسب لايربط منطقياً بين الاسم والمحتوى . فاسم الصندوق CAT (أى قطّ) مثلاً لايعنى بتاتاً أن الحاسب لايتوقع تخزين اسم قط بداخله . فصحة الموافقة بين المحتوى والاسم هى بالكلية من مهمة المبرمج، لذلك فإن من النصائح العامة فى مجال تسمية الصناديق أن يتوافق معنى اسم الصندوق مع محتواه، فمثلاً اسم «كمية» سيحتوى على كمية البضاعة المطلوبة، اسم «الرصيد» سيحتوى على الرصيد القابل للدفع من الزبون، ... وهكذا .

ومن الممكن أن يشير الاسم الواحد إلى صندوق واحد أو أكثر، وإذا أشار الاسم إلى أكثر من صندوق، فقد تكون هذه الصناديق مرتبة وفق قائمة ذات بعد واحد، مثل مجموعة الأسماء التابعة لمؤسسة معينة، فاسم المؤسسة هو اسم مجموعة الصناديق، وتحتوى الصناديق المنفردة ــ كل واحدة منها ــ على اسم لشخص معين . وقد يكون ترتيب الصناديق المتعددة لاسم واحد وفق جدول ذى بعدين، مثل المبيعات الأسبوعية من (الشاورما) لثلاثة فروع موزعة فى مدينة الرياض .

وتظهر هنا مشكلة فى كيفية تخصيص وتحديد أحادى الصناديق التى تشترك فى اسم واحد، وعولجت هذه المشكلة عن طريق إعطاء أرقام لأحادى الصناديق تتبع الاسم المشترك لها ومحاطة بقوسين، فإذا كان الاسم المشترك للصناديق ذات البعد الواحد هو «معهد» فإن الصندوق الأول يرمز إليه «معهد (١)» ... وهكذا و يعكس الرقم التالى هذا الأسلوب :

اسم مجموعة الصناديق : «معهد»

رقم الصندوق	١	٢	٣	٤	٥	٦	....

أما إذا كان ترتيب البيانات ذا بعدين، فيستخدم رقمان بينهما فاصلة «،» حيث يمثل كل رقم بعداً محدداً . وأقرب مثال لأذهان القراء لتمثيل هذه الحالة هى فى تسمية أرقام الغرف فى الفنادق ذات الطوابق المتعددة، حيث يتكون رقم الغرفة من جزأين الأول لرقم الطابق، والثانى لرقم الغرفة فى ذلك الطابق . ونلاحظ اشتراك الغرف فى الطابق الواحد فى رقم الطابق أفقياً ، واشتراك الغرفة ذات الموقع الواحد فى الطوابق المختلفة فى رقم الغرفة عمودياً . والاختلاف بسيط بين هذا الأسلوب فى تسمية الغرف، و بين الأسلوب الذى يجب اتباعه مع الحاسب من قبل المبرمج . ففى الحاسب يجب البدء من الطابق العلوى إلى الطابق السفلى، أى بعكس الاتجاه فى مثل الفندق، و يعكس الرسم التالى (صفحة ٩٧) المثل السابق حول المبيعات.

كما يلاحظ أن رقمى أحادى الصناديق، الأفقى والعمودى، تعزل بينهما الفاصلة «،»، و بناء على ذلك فإن شاورما الرياض (٢، ٢) يعكس مبيعات الأسبوع الثانى لفرع العليا، أى ١٠١ .

اسم الصندوق «شاورما الرياض»

الفرع / الأسبوع	البطحاء	العليا	أم الحمام
١	١،١ ٥٥	٢،١ ٦٢	٣،١ ٣٥
٢	١،٢ ٦٨	٢،٢ ١٠١	٣،٢ ٥٣
٣	١،٣ ٧٢	٢،٣ ٨٠	٣،٣ ٣١
٤	١،٤ ٧٩	٢،٤ ٧٥	٣،٤ ٨١

وأخيراً يجب التفريق بين اسم الصندوق ومحتواه، فالاسم ثابت لايتغير، فى حين أن محتواه ــ رقم المبيعات (١٠١) فى هذا المثال ــ يتغير من شهر لآخر.

أصناف البيانات :

تختلف طبيعة البيانات المخزنة فى هذه الصناديق باختلاف الهدف من حفظها وطريقة الاستعمال.

الهدف من الحفظ : إن الهدف من الحفظ قد يكون حسابياً أو غير حسابى، فالبيانات الحسابية تستخدم فى المعادلات الجبرية والمقارنات، أما غير الحسابية فيتعامل معها الحاسب كحروف متتابعة من الممكن أن تطبع كما هى أو تستخدم فى المقارنة فيما بينها، كفرز قائمة الأسماء حسب ترتيب الحروف الهجائية.

فالصندوق الحسابى يجب أن يحتوى على أرقام فقط دون حروف الهجاء والحروف الخاصة، أما الصندوق غير الحسابى فيمكن أن يحتوى على خليط من نوع أو أكثر من الحروف الرقمية والهجائية والخاصة.

وكثيراً ما تحتوى قواعد لغة البرمجة فى إطلاق الأسماء على الصناديق — على قاعدة محددة تميز محتويات البيانات كإضافة إشارة «$» فى نهاية اسم الصندوق للدلالة على أن محتوى الصندوق هو غير حسابى .

طريقة الاستعمال : إن البيانات المخزنة فى الصناديق إما أن تكون ثابتة أو متغيرة ، فالثابت من البيانات لا تتغير قيمته عند تشغيل البرنامج من فترة لأخرى . وهذا الثابت قد يكون حسابياً أو غير حسابى كذلك ، فالحسابى منه قد يخزن فى صندوق ذى اسم معين ، أو يثبت فى البرنامج كرقم محدد . ومن الأمثلة البسيطة على الثابت الحسابى هو الرقم «٢» الذى يستخدم لإيجاد مساحة المثلث بعد ضرب القاعدة فى الارتفاع . فأمامنا اتباع لإحدى الحالتين التاليتين :

الحالة الأولى : المساحة = $\frac{\text{القاعدة} \times \text{الارتفاع}}{٢}$

الحالة الثانية : ١ . القسمة = ٢

٢ . المساحة = $\frac{\text{القاعدة} \times \text{الارتفاع}}{\text{القسمة}}$

فيكون «القسمة» هو الرقم (٢) والذى لابد من تخزينه ، بالإضافة إلى قيمتى القاعدة والارتفاع ، وذلك قبل التمكن من إيجاد المساحة . وكذلك البيانات المتغيرة قد تكون حسابية أو غير حسابية . فالحسابية منها تتغير من مرحلة تشغيلية إلى أخرى . ففى المثال السابق يكون الصندوقان «القاعدة» و «الارتفاع» محتويين على بيانات متغيرة ، لذلك يطلق عليها الأسماء المتغيرة (Variable names) فى اصطلاح لغات البرمجة .

فعند تشغيل البرنامج للمرة الأولى لحساب مساحة مثلث ذى أبعاد محددة ، تخزن هذه الأبعاد فى الصندوقين المذكورين ، وإذا أراد شخص آخر إيجاد مساحة مثلث آخر ذى أبعاد مختلفة ، يتم استخدام نفس الصندوقين لتخزين الأبعاد الجديدة ... وهكذا .

أما الثابت والمتغيرمن البيانات غيرالحسابية فطريقة استعمالهما تشبه تلك الطريقة الخاصة بالحسابية منها ، فالبيانات المتغيرة غيرالحسابية تتبدل قيمتها من عملية تشغيل لأخرى ، فأسماء الموظفين تخزن كل شهر مرة فى الصناديق ذات الأسماء المتغيرة غير الحسابية ، فقد يكون هناك موظفون جدد قد أضيفوا إلى القائمة ، أو موظفون قد أحيلوا إلى التقاعد قد حذفوا من القائمة ... وهكذا .

أما البيانات الثابتة غيرالحسابية (Literals) فتعامل من قبل الحاسب وفق نصها الحرفى دون تحديد لطبيعة البيانات المخزنة ، فاسم الشركة الذى سيطبع على التقرير الشهرى لمبيعات «شاورما الرياض» ثابت لايتغيرمع التشغيل المتعدد على فترات زمنية ، وكذلك عنوانها إذا لزم الأمر .

وحكم الثابت غيرالحسابى مثل الثابت الحسابى ، فقد تخزن البيانات غيرالحسابية فى صندوق ذى اسم محدد ، فاسم الصندوق «شركة» قد يحتوى على «شاورما الرياض» ، وقد نطبع كلمتى «شاورما الرياض» مباشرة من البرنامج عن طريق وضعهما بين الإشارتين (« ») ، كما هومكتوب فى الأسطر السابقة .

ويمثل الجدول التالى تلخيصاً لأصناف البيانات حسب الهدف من الحفظ وطريقة الاستعمال .

أصناف البيانات حسب طبيعة الاستعمال مع أمثلة

الهدف من الحفظ	طريقة الاستعمال (طبيعة البيانات)	
(طبيعة المعالجة)	— متغيرة — تخزن فى صناديق لها أسماء	— ثابتة تظهر فى البرنامج
— حسابية .	الطول والعرض والكمية .	٢ ، ٣ ، ط
— غيرحسابية .	الاسم ووصف المخزون والعنوان	«شاورما الرياض» «شركة داود للخزف»

أصناف العبارات التوضيحية :

إن مجرد إظهار البيانات ونتائج المعالجات على وسائط الإخراج المرئية من قبل المستفيدين مثل الشاشات والورق، لن يؤدى الغرض المنشود منها . فهذه البيانات غالباً ما تتكون من أرقام تمثل أشياء متباينة بحاجة إلى تعريف وتحديد . وقد تعرضنا فى الجزء السابق لأحد هذه الأنواع من التعريف وهو الثابت غير الحسابى .

بالإضافة إلى ذلك، فإن تشغيل البرنامج قد يتطلب تعليمات توضيحية لما يجب أن يفعله المستفيد لتشغيل البرنامج بنجاح، ودون توقف أو انقطاع .

ونلاحظ أن هذه العبارات التوضيحية ثابتة لا تتغير مع التشغيل المتكرر للبرنامج، ومن أنواعها :

أ ــ **العناوين (Labels)** : يوضح العنوان مجموعة من البيانات التى ستدخل من قبل المستفيد، أو ستطبع من قبل البرنامج، فعبارة «المعلومات الشخصية عن الموظف» التى تظهر فى أعلى الشاشة تفيد بأن البيانات التى ستدخل/ ستطبع لها صفة مشتركة .

وغالباً مايستعمل العنوان لتعريف مجموعة من البيانات لها خاصية واحدة، ومرتبة إما أفقياً أو عمودياً . و يعطى الشكل التالى مثالاً لاستخدامات العناوين :

المبيعات الأسبوعية لفروع
شركة شاورما الرياض لشهر ربيع الأول ١٤٠٧ هـ

	الفروع		
الأسبوع	البطحاء	العليا	أم الحمام
الأول			
الثانى			
الثالث			
الرابع			

فكلمة «الأول» تصنف مبيعات الأسبوع الأول للأفرع الثلاثة، في حين أن كلمة «البطحاء» تصنف مبيعات فرع البطحاء للأسابيع الأربعة في شهر ربيع الأول لعام ١٤٠٧ هـ ... وهكذا .

ب ــ **تعريف البيانات (Data Definitions) :** في بعض الحالات يستدعي تعريف أحادي البيانات بدلاً من مجموعة من البيانات . فكلمة «الاسم» تدل على أن اسماً سيظهر عقبها أو تحتها ، و يعطي المثال التالي توضيحاً لهذا الاستخدام :

المعلومات الشخصية عن الموظف

الاسم :	العمر :
تاريخ الميلاد :	تاريخ التعيين :
عدد الأولاد :	

وقد يستخدم كلا المثالين السابقين لإدخال البيانات أيضاً .

جـ ــ **المذكرات (Comments) :** تستخدم المذكرات بشكل رئيسي لتيسير وتسهيل عملية التخاطب بين المستفيد والبرنامج ، وغالباً مايكون المستفيد قليل أو معدوم الخبرة في مجال الحاسب . وقد يؤدي أية إعاقة أو توقف للبرنامج أثناء التشغيل إلى فقدانه الثقة في قدرات الحاسب . وتأخذ المذكرات أنواعاً متعددة :

فمذكرة الأوامر تعطي توجيهات لمستخدم النظام ، كمدخل البيانات ، وذلك حتى تتم عملية إدخال البيانات بالطريقة السليمة خالية من الأخطاء ، فعبارة :

«أدخل الاسم مكوناً من ١٥ حرفاً على الأكثر» تدخل ضمن هذا الصنف .

وكذلك تظهر عبارات الأوامر في شاشات الاختيارات والتي تحدد مسار التنفيذ في البرنامج كالشاشة التالية :

شاشة الاختيارات الرئيسية
١ ـــ إدخال بيانات الموظف الشخصية .
٢ ـــ اطبع بيانات الموظف الشخصية .
٣ ـــ الخروج من النظام .
أدخل الاختيار المطلوب :

تحدد ثلاثة اختيارات على المستخدم أن يطبع أحدها ، ومن ثم يتم التشعب فى البرنامج لتحقيق المطلوب .

ومذكرة الأخطاء تستخدم لتنبيه مستخدم البرنامج إلى أن خطأً معيناً قد وقع أثناء تشغيل البرنامج ، فالعبارة :

«الاسم يشتمل على أرقام ، أدخل الاسم مرة أخرى خالية من الأ رقام من فضلك»

فيها تنبيه للخطأ وتوضيح لما يجب فعله لتصحيحه . وقد تظهر هذه المذكرات مباشرة على الشاشة إذا كان التشغيل للبرنامج فى الوضع التبادلى . وفى بعض الأحيان قد يتم تحضير قائمة بالأخطاء وطبعها جميعاً مرة واحدة ، ومثال ذلك ، تحضير قائمة بالقطع التى تكون أرصدتها سالبة القيمة ، أو قائمة بأسماء الموظفين الذين لا تتوافق تواريخ تعيينهم مع تواريخ ميلادهم .

والملاحظات لها وظيفة إعلامية حيث توضح للمستخدم حالات معينة يكون المستخدم بدونها حائراً فى تفسير ما يجرى ، فالعبارة :

«أرجو الانتظار قليلاً لحين الانتهاء من تلبية طلبك»

تزيح عن ذهن المستخدم الارتباك أثناء الانتظار لفترة وجيزة يحار معها فيما يعمل ، وكذلك العبارة :

«هل تريد تعديل السجل كما طلبت أدخل نعم أو لا»

تحدد مرحلة فى عملية التحديث يطمئن معها المستخدم إلى الحاسب .

كذلك قد تظهر الملاحظات فى أسفل وثيقة ما كتمييز عميل أو زبون عن آخر .
فعبارة : «شكراً على مواظبتكم على دفع أرصدة حسابكم فى وقتها»
لها مفعول إيجابى فى نفس الزبون وقد تشجعه على الاستمرار فى عادته تلك .

المعالجات الأساسية التى يقوم بها الحاسب :

كما مر ذكره، فإن المعالجات التى يقوم بها الحاسب تتكون من جميع العمليات الحسابية والمنطقية، ولابد من الإشارة هنا إلى أن الحاسب يقوم بعملية حسابية واحدة أو عملية منطقية واحدة فى نفس الوقت، إذ لا يستطيع الحاسب أن يجمع بين عمليتين منطقيتين أو بين عملية حسابية وأخرى منطقية . وسنتعرض الآن لهذه المعالجات :

أ ــ **العمليات الحسابية وأولويتها :** بالإضافة إلى العمليات الحسابية الأساسية من جمع وضرب وقسمة وطرح، فإن الحاسب يستطيع تمييز عمليات الأسس والجذور، ومن الممكن استخدام الأقواس لتحديد تسلسل المعالجات الحسابية فى معادلة جبرية طويلة .

ومن الأمثلة على المعادلات الجبرية ذات المتغيرات المتعددة والتى يمكن للحاسب القيام بها :

أ ــ $C = A + B$

ب ــ $D = A + B - D + 3$

جـ ــ $E = A \div B + C$

د ــ $F = (A - B) \div 2 - C$

هـ ــ $X = (A)^3 - (B + C) - 10$

و ــ $Y = [A \times B - (B)^2 \ X \ 3] + 10$

فبالنسبة لتسلسل العمليات الحسابية فى العبارات السابقة فهو واضح فى الثلاث الأول، وفى العبارة (د) يتم طرح B من A أولاً، ثم يقسم الناتج على ٢، وأخيراً يطرح C من الناتج الأخير. أما فى المثال هـ، فيتم رفع المتغير A للقوة ٣ ثم تجمع B و C ويطرح ناتج ذلك من القيمة الناتجة عن الأس. وأخيراً يطرح ١٠ من الناتج الأخير للوصول للناتج النهائى. وأخيرا فى العبارة (و) يتم ضرب الناتج من رفع B للأس ٢ فى العدد ٣، وتطرح نتيجة هذه العملية من ناتج ضرب A فى B، وأخيرا يضاف للناتج ١٠.

وهذا التسلسل فى جزئيات العمليات الحسابية لايتأثر بمستوى سهولة وصعوبة المعادلة الجبرية.

ولنأخذ المثالين التاليين :

أ) ٢ × ٢ - ١ ب) ١٤٩,٩٩ × ٢ - ١

فبالرغم من أن الإنسان يقوم بالعملية الحسابية الأولى بسرعة ودون الحاجة إلى الرجوع للورقة والقلم، كما قد يستدعى الحال فى المثال الثانى، فإن الحاسب يتبع نفس الخطوات فى كلا المثالين. إذ يجب أن تخزن نتيجة الضرب أولاً، ومن ثم يتم الطرح.

وبالإضافة إلى تسلسل العمليات الحسابية فى المعادلة الجبرية، فإن هناك معايير عامة تنطبق على جميع العبارات الحسابية بغض النظر عن لغة البرمجة المستخدمة :

١) وجود متغير واحد فقط إلى يسار إشارة «=» وذلك يرمز إلى اسم الصندوق الذى سيتم فيه تخزين الناتج من العملية الحسابية.

٢) استخدام إشارة «=» هنا ليس فى معناها الجبرى المجرد، فإشارة «=» تعنى «إعطاء» أو «إسناد» القيمة النهائية للعمليات الحسابية إلى المتغير الظاهر على يسار الإشارة «=».

فمعنى العبارة التالية إذن :

, C = C + D

هـو : أضـف الـقـيـمـة الـسابقة المخزنة فى C إلى القيمة المخزنة فى D وضع القيمة الناتجة فى C مرة أخرى لتمثيل القيمة الجديدة لها .

٣) استخدام رموز خاصة فى لغة البرمجة لتمثيل العمليات الحسابية ، وذلك كالتالى :

الإشارة الجبرية	الرمز الجبرى المقابل
+ الجمع	+
− الطرح	−
÷ القسمة	/
× الضرب	*
x^3 الأس	** أو ↑

ب ــ **العمليات المنطقية :** يتضمن المجالان التجارى والحكومى الكثير من التطبيقات التى تتطلب إجراء مقارنات بين بيانات مختلفة ، إذ لا يخلو أى تطبيق منها . وتحدثنا فى الفصل الثانى عن التشعب المبنى على مقارنة معينة وتمثيلها بالرمز التالى ◇ فى نمط البرمجة التركيبية ، وسنورد هنا مثالين آخرين :

المثال الأول : إذا تصورنا الكبينات التى تحتوى أدراجها على ملفات الموظفين لجهة حكومية معينة ، فقد نرغب فى الحصول على المعلومات التالية :

أ ــ عدد الموظفين الذين تزيد أعمارهم على ٤٥ سنة .

ب ــ أسماء الموظفين الذين سيتقاعدون خلال السنة القادمة .

جـ ــ عدد الموظفين ونسبهم ، حسب توزيعهم بين الجنسيات المختلفة .

د ــ أسماء الموظفين التابعين لدائرة معينة موزعين حسب فئات الأعمار التالية :
أقل من ٢٠ ، من ٢٠ إلى أقل من ٣٠ ، من ٣٠ إلى أقل من ٤٠ ، من ٤٠ فأكثر .

هـ ــ عدد الوظائف الشاغرة فى السنة المالية .

و ــ أسماء الموظفين التابعين لدائرة معينة وتزيد أعمارهم على ٤٠ سنة .

ز ــ أسماء الموظفين التابعين للدائرة المالية أو للدائرة التجارية أو لدائرة شؤون الموظفين .

ح ــ أسماء الموظفين التابعين للدائرة المالية أو للدائرة التجارية وأعمارهم تزيد على ٤٥ سنة .

المثال الثانى :

إذا تصورنا بطاقات المخزون تحتوى على معلومات عن الأنواع المختلفة للقطع المخزونة فى المستودع ، لأ ردنا الحصول على المعلومات التالية :

أ) قائمة برموز القطع وأوصافها ، والكمية الحالية المخزنة الناقصة عن مستوى محدد مسبقاً .

ب) قائمة برموز القطع وأوصاف القطع ، التى لايتم الطلب عليها باستمرار .

جـ) مجموع قيم القطع المنصرفة ، موزعة حسب المناطق الجغرافية الشرقية والوسطى والغربية .

د) أسماء القطع المنصرفة للدائرة المالية ، أو لدائرة الإنتاج وتزيد قيمتها على ١٠٠٠ ريال .

هـ) أسماء القطع المنصرفة للدائرة الإدارية وتزيد قيمتها على ٢٠٠٠ ريال ، أو المنصرفة لدائرة شؤون الموظفين وتزيد قيمتها على ١٥٠٠ ريال .

و) أرقام وأسماء القطع التى يزيد سعر الواحدة منها على ٥٠٠ ريال ، موزعة حسب طبيعة وحدة البيع : بالقطعة الواحدة ، بـــ «الدرزن» ، بالرزمة .

إن إجراء المقارنات المذكورة فى المثالين السابقين يدو ياً يتطلب الحفاظ على البيانات الضرورية ، والمناسبة لتحقيقها فى بطاقات الموظفين و بطاقات المخزون ، وعند

استخدام الحاسب الآلى، يتم تخزين البيانات فى سجل لكل موظف ولكل قطعة فى المخزون . ومن الطبيعى أن تحتوى هذه السجلات على : بيانات أخرى قد لا تستخدم فى المقارنات المذكورة آنفاً .

تصنيف المقارنات : تستخدم المقارنات فى عمليات التصنيف للسجلات المخزنة فى الحاسب أو للاستخلاص الجزئى لبعض منها، وكما مر ذكره فى بداية هذا الفصل، فإن البيانات التى ستستخدم فى المقارنات تخزن فى أسماء متغيرات (صناديق) و يطلق على كل واحدة من هذه البيانات حقل، وتسهيلاً للشرح فى هذا الجزء، فإن أسماء هذه المتغيرات ستكون عبارات باللغة العربية، أما فى حالة البرمجة بإحدى اللغات، فإن هذه الأسماء يجب أن تخضع لقوانين تلك اللغة .

ويمكن تصنيف المقارنات إلى صنفين رئيسيين :

الأول : البسيط، و يتم فيه المقارنة بين قيمتين من البيانات فقط . وفى العادة تكون إحدى القيمتين مخزنة فى اسم متغير ــ العمر مثلا ــ أما القيمة الأخرى فقد تكون بيانات ثابتة سواء حسابية رقمية أو غير حسابية حرفية أو بيانات متغيرة، ومن الأمثلة على هذا الصنف : العمر > ٤٥ حيث يخزن العمر لكل موظف و يسترجع بعد ذلك لمقارنته بالرقم «٤٥» .

والكمية > الحد الأدنى

حيث يخزن الكمية والحد الأدنى لكل قطعة . ففى هذه الحالة يتم وضع الأجزاء المكونة من مقارنتين أو أكثر بين قوسين، و يتم معاملة هذه الأجزاء كوحدة واحدة لها نتيجة خاصة بها، ومن ثم نقارن نتائج مقارنات الأجزاء لتحديد نتيجة المقارنة الكلية، ومن الأمثلة على ذلك :

ــ (اسم الدائرة «=» المالية أو اسم الدائرة «=» شؤون الموظفين) والعمر > ٤٥ .

ـ (اسم الدائرة = المالية والقيمة > ٢٠٠٠) أو (اسم الدائرة = «الموظفين» والقيمة < ١٥٠٠) .

ففى المثال الأول يقوم الحاسب بإجراء المقارنتين الموجودتين داخل القوسين وتحديد النتيجة لهذا الجزء، ومن ثم تربط هذه النتيجة بنتيجة المقارنة الأخيرة . وفى المقال الثانى يتم تحديد نتيجة المقارنات المجزأة والمحصورة بين الأقواس، بحيث تحدد النتيجة النهائية للعبارة بناء على النتيجتين الجزئيتين .

كيفية تنفيذ عبارة المقارنات : إن الهدف من المقارنة هو تحديد اختيار أحد مسارين بناء على تحقق شرط المقارنة أو عدمه :

ـ رمز المنطقة = ١

حيث يخزن رمز المنطقة و يقارن مع «١» الذى يشير إلى المنطقة الوسطى .

الثانى : المركب، حيث تحتوى العبارة الواحدة على مقارنتين، و ينطبق على الصنف ما انطبق من القواعد على الصنف الأول من حيث طبيعة البيانات المقارنة، ويمكن تقسيم هذا الصنف إلى صنفين فرعيين :

١ ـ **المركب العادى :** والذى يحتوى على المقارنات دون اعتبار لتحديد تسلسل تنفيذ أجزائها (مقارناتها) . ومن الأمثلة على هذا النوع :

ـ اسم الدائرة = «المالية» والعمر > ٤٥

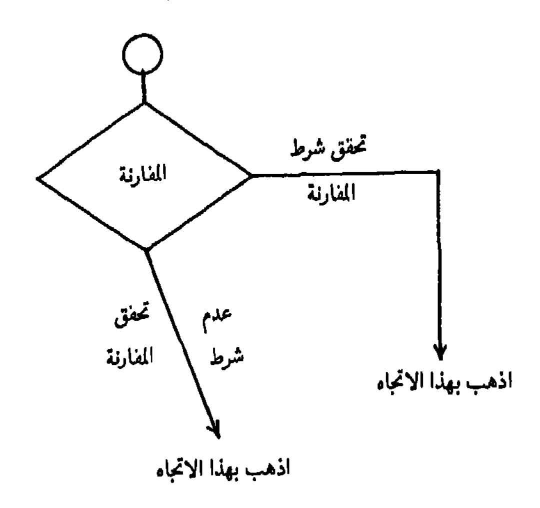
ـ اسم الدائرة = «المالية» والعمر > ٤٥ والحالة الاجتماعية = «أعزب» .

ـ اسم الدائرة = «المالية» أو اسم الدائرة = «الإدارية» أو اسم الدائرة = «الموظفين» .

ـ سعر الوحدة > ٥٠٠ أو اسم الدائرة المتسلّمة = «المالية» .

٢ ـ **المركب المجزأ :** والذى يحتوى على عدة مقارنات بحيث تجمع مقارنتان أو أكثر فى جزء واحد، ومن الممكن أن تحتوى عبارة المقارنة على جزأين أو أكثر.

ويمثل الشكل التالى كيفية إجراء المقابلة :



ويمكن تبادل مكان الاتجاهين فى رمز المقارنة ◇ دون أن يؤثر ذلك على سير تنفيذ الخطوات الأخرى .

الفرق بين «و» و «أو» : عند إجراء المقارنة ، فإن المقارنات المركبة العادية المربوطة بـ «و» يجب أن تتحقق جميعها ، حيث يتحقق الشرط الكلى للعبارة الشرطية ، فمثلاً ، تؤدى مقارنة اسم الدائرة «و» العمر إلى إنتاج قائمة بالأسماء التى يتحقق فيها الشرطان : أن تكون تابعة للدائرة المالية والعمر أكبر من ٤٥ ، وكذلك إذا أضيف شرط المقارنة مع الحالة الاجتماعية .

أما عند استخدام حرف «أو» فإن تحقق الشرط الكلى للعبارة يتم إذا تحققت أى من الشروط المربوطة بـ «أو» . فالمقارنات عن الدائرة المالية «أو» الدائرة الإدارية «أو» دائرة الموظفين تؤدى إلى إنتاج قائمة بأسماء الموظفين التابعين لأى من هذه الدوائر .

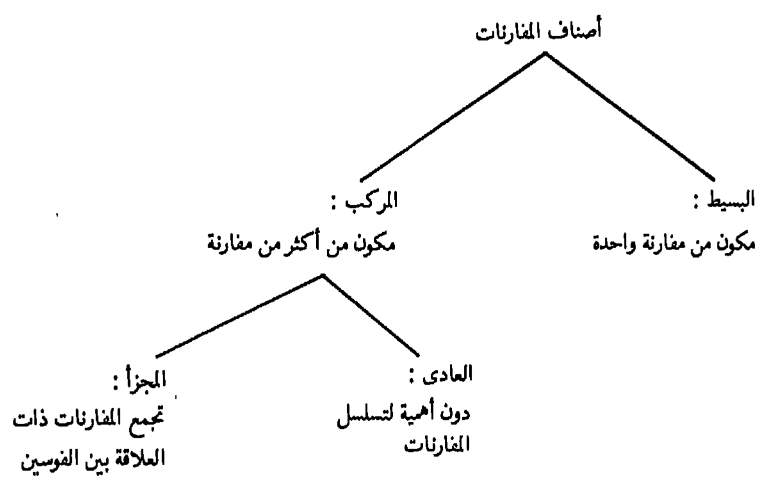
و يتبع نفس الأسلوب فى إجراء المقارنات المتضمنة أقواساً ، فالمقارنة عن الدائرة المالية أو الدائرة الإدارية المحصورة بين قوسين تؤدى إلى اختيار أسماء الموظفين فى هاتين

الدائرتين، ومن ثم فإن مقارنة العمر تؤدى إلى حصر القائمة فقط فى الموظفين الذين تزيد أعمارهم على ٤٥ عاماً .

أما المقارنة عن الدائرة المالية والقيمة أو دائرة الموظفين والقيمة، فينتج عنها أسماء القطع التى صرفت للدائرة المالية، وقيمتها تزيد على ٢٠٠٠، والقطع التى صرفت لدائرة الموظفين، وتزيد قيمتها على ١٥٠٠ .

لاحظ أن صياغة عبارة المقارنة بـ «و» عند استخدام نفس الاسم المتغير يستدعى تكراره، ومثال ذلك : إذا كان العمر 〉 ٢٠ والعمر = أكبر من ٣٠، فنكرر استخدام العمر مرتين وليس مرة واحدة .

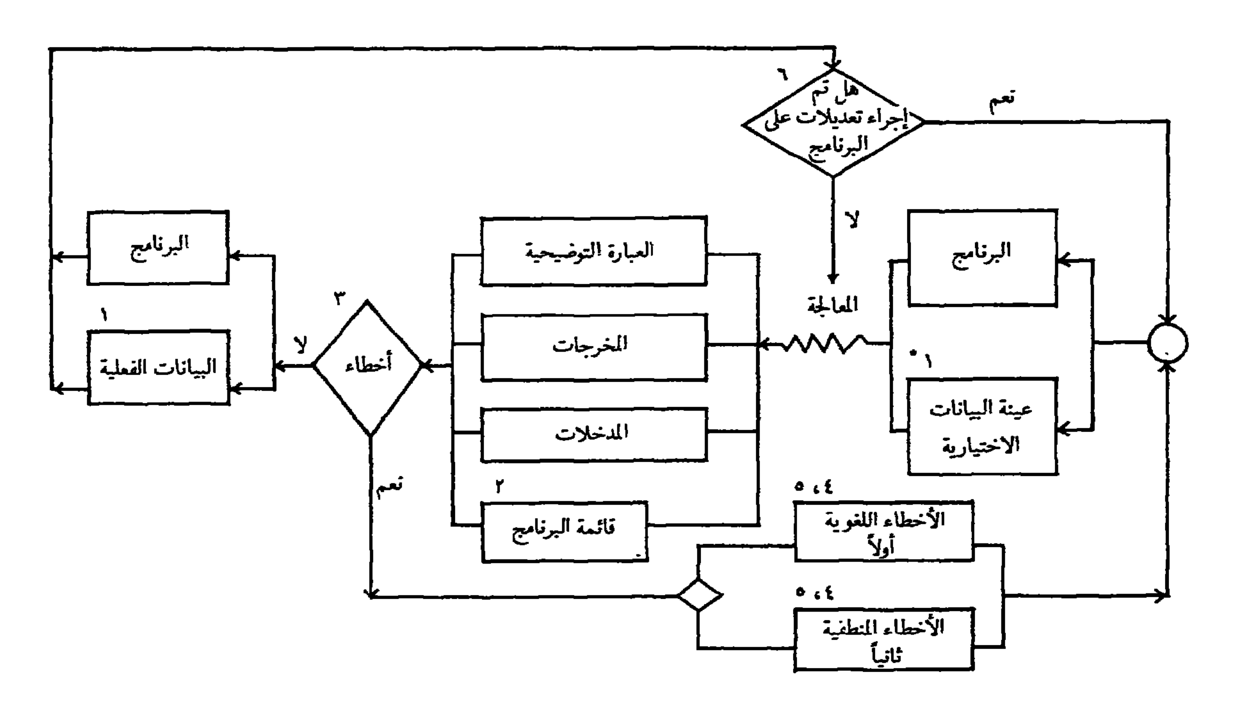
ويمثل الشكل الشجرى التالى تلخيصاً لأصناف المقارنات :



طبيعة التخاطب مع الحاسب :.

بعد أن تحدثنا عن طبيعة البيانات وحالات معالجتها، لابد من شرح الترابط بين العمليات التى تصاحب التخاطب مع الحاسب الآلى، فالهدف الرئيسى لهذا التخاطب هو معالجة البيانات المدخلة لإنتاج المعلومات، و يتطلب ذلك تحديد خطوات المعالجة فى برنامج مكتوب بإحدى لغات البرمجة . وسنستعين بالشكل التالى لشرح طبيعة التخاطب مع الحاسب عند تطوير برنامج معين .

شكل (٤ ــ ١) أسلوب التخاطب مع الحاسب عند تطوير برنامج معين .



* يشير الرقم فوق الصندوق إلى رقم الملاحظة التى تتبع فى الصفحات التالية .

نلاحظ من الرسم ما يلى :

١ ـ بعد كتابة البرنامج يتم تنفيذه باستخدام عينة من البيانات ، و يتم اختيار هذه العينة بحيث يسهل الوصول إلى النتائج يدو ياً ، وذلك من أجل مقارنتها مع النتائج التى سيظهرها الحاسب ، و بحيث تغطى كافة الاحتمالات والمسارات التى يمكن أن تتخذها المعالجة .

فعند مقارنة ساعات العمل مع الرقم ٤٠ ، الذى يمثل العدد الطبيعى لساعات العمل الأسبوعية ، وذلك من أجل معرفة وجود ساعات إضافية من عدمه ، لابد أن تحتوى البيانات الاختيارية على أرقام لعدد الساعات الأسبوعية = وأكبر من وأصغر من ٤٠ . أما كون أجرة الساعة ذات أرقام عشرية ، فلا يعنى بالضرورة أن تحتوى بيانات الاختيار على أرقام مثل ٥٠,٩٩ أو٤٩,٣٨ ، إذ أن هذه الأ رقام لا تؤثر على منطق تسلسل خطوات البرنامج .

و بعد التأكد من صحة النتائج ، يتم تنفيد البرنامج باستخدام البيانات الفعلية .

٢ ـ التفريق بين قائمة البرنامج الذى يحتوى على عبارات البرمجة حسب ما تتم طباعته من قبل المبرمج ، و بين النتائج التى تظهر نتيجة لتنفيذ البرنامج . ففى أى مرحلة من مراحل التخاطب مع الحاسب ، يمكن الحصول على قائمة عبارات البرنامج ، من أجل عمليات التدقيق والمراجعة بشكل أساسى ، فى البداية ، ومن أجل إرفاقها مع نتائج التنفيذ عند التأكد من تكامل وصحة المعلومات المخرجة .

٣ ـ يمكن التفريق بين نوعين من الأخطاء التى من الممكن أن تقع عند محاولة تنفيذ البرنامج : الأخطاء اللغوية Syntax Errors والأخطاء المنطقية Logical Errors . فالأخطاء اللغوية هى التى لها علاقة باللغة ، وتنتج عن مخالفة قواعدها . ففى لغة بيسك مثلاً ، يمكن ملاحظة الأخطاء اللغوية التالية :

ــ (خطأ فى تهجى PRINT) 10 PRNT

ــ (إسناد بيانات حرفية لمتغير رقمى) 20 A = «Good Morning»

وهناك أخطاء أخرى يمكن الرجوع إليها فى دليل اللغة المستخدم . أما الأخطاء المنطقية ، فلها علاقة بخطوات سير البرنامج ، كوصف خطوة تابعة لأخرى كان يجب أن تسبقها ، كمحاولة حساب الأجر الإضافى قبل قراءة عدد الساعات أو قبل التدقيق فيها . والأخطاء اللغوية تكتشف من قبل الحاسب عند تسلمه تعليمات اللغة . أما الأخطاء المنطقية فعلى المبرمج اكتشافها عن طريق مراجعة خطوات البرنامج ونتائج البيانات الاختيارية .

٤ ــ ميقات اكتشاف الأخطاء يتم حسب المرحلتين اللتين تمر بهما عملية التخاطب مع الحاسب :

- مرحلة ترجمة تعليمات اللغة إلى اللغة التى يفهمها الحاسب ، و يطلق على البرنامج الناتج عن هذه المرحلة «البرنامج الهدف» «Object Program» .
- مرحلة تنفيذ التعليمات المترجمة والتى تؤدى إلى قيام البرنامج الهدف بتنفيذ الخطوات الحسابية والمنطقية المنصوص عليها فى البرنامج . ففى المرحلة الأولى يتم اكتشاف الأخطاء اللغوية ، فى حين يتم اكتشاف الأخطاء المنطقية فى المرحلة الثانية .

ويمكن تحقيق المرحلة الأولى بأحد أسلوبين :

ــ أسلوب الترجمة الفورية للتعليمات (المترجم) Interpreter

ــ أسلوب التأليف (المؤلف) Compiler

ففى الأسلوب الأول ، تتم ترجمة أسطر البرنامج سطراً سطراً حال إرسالها إلى الحاسب ، و يتم اكتشاف الأخطاء اللغوية لكل سطر على حدة . و يستخدم هذا الأسلوب بشكل خاص مع لغة بيسك ، وإن لم يكن هو الأسلوب الوحيد .

أما فى أسلوب التأليف، فتتم قراءة البرنامج كاملاً، ومن ثم اكتشاف الأخطاء اللغوية، بعد التدقيق فى تجانس قواعد اللغة المستخدمة فى البرنامج ككل متكامل . ومن الممكن استخدام هذا الأسلوب مع لغة بيسك باستخدام مؤلف اللغة بدلاً من استخدام اللغة بالأسلوب العادى المتعارف عليه، أى الأسلوب الأول .

وقد تم شرح ميقات اكتشاف الأخطاء بشكل مبسط دون الخوض فى الكثير من التفصيلات .

٥ ــ بناءً على المناقشة السابقة، نجد أن الرسم السابق يناسب استخدام لغة بيسك مع المترجم، أما فى حالة استخدام أسلوب التأليف، فيتم فى العادة تنقية البرنامج من الأخطاء اللغوية قدر الإمكان، و بعد ذلك يتم دمج البرنامج مع عينة الاختبارات وتنفيذه للتأكد من خلوه من الأخطاء المنطقية .

٦ ــ إنه، وإن كان قد تم تنفيذ البرنامج بنجاح حسب المواصفات والمتطلبات الأولية، فلا بد وأن تطرأ ظروف ومتطلبات جديدة تتطلب إجراء التغييرات والتعديلات على البرنامج، ومن ثم يتم العودة إلى نقطة البداية والمرور فى نفس المراحل المتتالية .

تمارين

١ ــ اشرح باختصار العمليات التى يمكن أن تتم على محتو يات الصناديق (المتغيرات) فى ذاكرة الحاسب الآلى.

٢ ــ بين كيفية تعامل الحاسب مع الجدولين التاليين :

أ ــ تخزين المبيعات اليومية لمدة أسبوع .

ب ــ تخزين المبيعات اليومية حسب موقعها من الأسابيع الأ ربعة فى كل شهر.

٣ ــ ماهى أصناف البيانات حسب طبيعة الاستعمال مع إعطاء مثالين لكل صنف؟

٤ ــ ماهى أصناف العبارات التوضيحية مع إعطاء نوعين لكل صنف؟

٥ ــ ماهو تسلسل تنفيذ العمليات الحسابية ؟

٦ ــ صنف المقارنات التالية :

أ ــ السن أكبر من ٦٠ .

ب ــ الدخل أقل من ٥٠٠٠ ريال .

جـ ــ الدارس من خارج منطقة الرياض و يسكن فى سكن المعهد .

د ــ الأستاذ يقوم بتدريس أكثر من عشر ساعات ويحضر للتدريس ليلاً و يعمل فى استشارة.

هـ ــ المدير ذو الراتب أكثر من عشرة آلاف ريال شهرياً ، و يعمل فى فرع الرياض، أو المدير ذو الراتب بين خمسة آلاف إلى ثمانية آلاف ريال و يعمل فى فرع جدة .

الجزء الثانى

أساسيات استخدام لغة بيسك

برمجة معالجات بسيطة.
برمجة معادلات بسيطة
مع عمليات إدخال.
الدوارة البسيطة
والتحكم بها.
تعليمات التشعب .
استخدام الدالات
فى العمليات الحسابية.
التحكم فى طبع البيانات
المخرجة والمعلومات.
المصفوفات ذات البعد
الواحد وتعليمات
الدوارة البسيطة.
المصفوفات ذات البعدين
ومكوناتها وتعليمات
الدوارة المتقدمة.

الفصل الخامس

برمجة معالجات بسيطة

حالات التخاطب مع الحاسب الآلى :

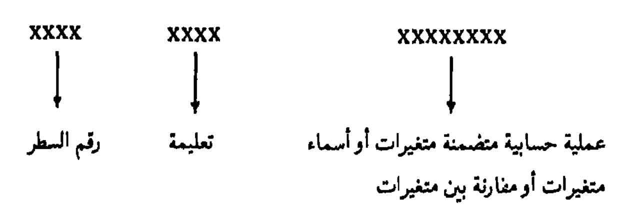
بعد تجهيز الحاسب والوصول به إلى مرحلة قبول تعليمات لغة بيسك ، يمكن التخاطب مع الحاسب الآلى ضمن الحالات التالية :

أ) **حالة الاستعداد :** يمكن طباعة أى تعليمة من تعليمات لغة بيسك دون كتابة رقم سطر ، فمثلا نستطيع أن نطبع التعليمات التالية :

```
LET A = 10
LET B = 5
LET C = A ¥ B
PRINT A,B,C
```

وقد يستخدم هذا الوضع فى بداية تعلم اللغة ؛ لأخذ فكرة عامة عن كيفية تخزين البيانات فى ذاكرة الحاسب وطباعة محتو ياتها ، بالإضافة إلى ذلك يمكن التوصل إلى معرفة محتو يات أسماء المتغيرات بعد تنفيذ البرنامج ، كما سنتعرض له بعد قليل .

ب) **حالة البرمجة :** نبدأ حالة البرمجة فور طباعة رقم السطر لعبارة بيسك ، و يبين الشكل التالى مكونات العبارة :



ويجب أن يكون رقم السطر بين 00001 إلى 99999 . أما التعليمة فهى من الكلمات ذات الدالة الخاصة لمترجم لغة بيسك (RESERVED WORD)، ويجب أن تظهر بأحرفها كاملة دون زيادة أو نقصان، وسيتم كتابة هذه العبارة فى الأشكال التوضيحية لها بالحرف الكبير، مثل PRINT، أما أسماء المتغيرات فسيتم استخدام الأحرف الصغيرة لها . وإذا تضمنت المقارنات رموزاً، مثل (و AND)، (أو ــ OR)، فستعامل مثل التعليمة . أما عند طباعة البرامج فنجد أن الأمر سيان فى كيفية طباعة التعليمات . فمثلاً :

ــ الشكل التوضيحى لعبارة اطبع PRINT

XXXXX	PRINT	X1,X2,X3
رقم السطر	تعليمة	أسماء المتغيرات

ــ طباعة تعليمة PRINT فى برنامج معين

```
10 PRINT N1,N2,N3
```

ولابد من الضغط على مفتاح العودة/ الإدخال بعد الانتهاء من طباعة كل عبارة .

جـ) **حالة التشغيل :** يتم الانتقال إلى حالة التشغيل عند طباعة كلمة RUN، ويكون ذلك بعد الانتهاء من طباعة برنامج معين إما لاختباره أو للحصول على النتائج المرجوة، كما هو فى المثال التالى :

```
10 LET A = 5
20 LET B = 7
30 LET C = A + B
40 PRINT A,B,C
RUN
       5       7       12
```

ويمكن بعد تنفيذ البرنامج التخاطب مع الحاسب من حالة الاستعداد . فمثلاً :بعد تنفيذ البرنامج السابق يمكن كتابة التعليمة التالية :

```
PRINT  A,B,C
```

وستظهر النتائج السابقة بعد الضغط على مفتاح العودة/ الإدخال .

معالجة عمليات حسابية بسيطة :

لنبدأ بشرح أسلوب برمجة الحاسب بلغة البيسك بمثال بسيط بدائى وذلك لأغراض التعليم والإيضاح .

مثال (٥ ــ ١) : من العمليات الحسابية الكثيرة التى يقوم بها بائع السجاد عند حساب المبلغ الإجمالى الذى سيدفعه الزبون : إيجاد مساحة السجاد المبيع.

الهدف هو إيجاد مساحة السجاد المبيع إذا علم الطول والعرض.

وخطوات الحل هى :

١ ــ الحصول على الطول والعرض .

٢ ــ حساب المساحة وفقاً لمعادلة المستطيل .

المساحة = الطول × العرض .

٣ ــ إيضاح النتائج .

وسنتعرض الآن للعبارات المطلوبة فى لغة البيسك ، والتى ستترجم هذه الخطوات حتى يستطيع الحاسب تنفيذها ، ونشير هنا إلى أننا سنبدأ من وضع البرمجة Program Mode والذى يتطلب إعطاء أرقام لأسطر العبارات فى لغة بيسك.

الخطوة الأولى ــ الحصول على البيانات المدخلة :

إن الحصول على الطول والعرض ــ المدخلات ــ اللازمة للخطوة التالية ــ المعالجة ــ يمكن التوصل إليه عن طريق أكثر من أسلوب واحد وبغض النظر عن

الأسلوب المختار، فإننا نعلم أن هذه الخطوة تتطلب تحديد صندوقين فى ذاكرة الحاسب لتخزين رقمى الطول والعرض بداخلهما ـــ متغيرين . وأسماء هذه المتغيرات الحسابية ـــ الصناديق ـــ تخضع لقوانين لغة بيسك ، كما هى موضحة فى الشكل التالى :

قواعد تسمية المتغيرات الحسابية :

١ ـــ يبدأ بحرف أبجدى A-Z

٢ ـــ لايبدأ برقم 0-9

٣ ـــ يمكن أن يكون الحرف الثانى إما أبجدياً أو رقمياً .

٤ ـــ لا تنتهى الأسماء بإشارة $

أسماء صحيحة A, B, BA, B1

أسماء غير صحيحة AC$,A $, A1 $,1A

٥ ـــ بعض الحاسبات تحدد عدد الحروف المستعملة باثنين أو ثلاثة أو ثلاثين .

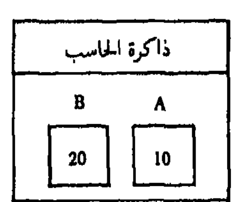
IBM	: ٣٠
RADIO- SHACK	: ٣
HP- 3000	: ٣٠
APPLE	: ٣

٦ ـــ لاتحتوى على فراغ Space أو «.»

فالأسماء التالية خاطئة AB, CA.1, A1 $

٧ ـــ يمكن استخدام % للدلالة على كون الرقم المخزن كاملاً لايحتوى على كسر عشرى، مثل A %

واستخدام أسماء المتغيرات الحسابية هذه يكون ضمن عبارات البيسك التى ستقوم بتخزين الأرقام بداخلها . ولكن النتيجة المرجوة موضحة فى الشكل التالى :



حيث يمثل الصندوق «A» العرض، والصندوق «B» الطول، والقيم ١٠، ٢٠ قدماً على التوالى* .

وللوصول إلى هذه الحالة سنتبع الآن واحداً من الأساليب المتاحة وهى استخدام عبارة الإسناد LET

عبارة الإسناد LET :

تستخدم هذه العبارة لتخزين البيانات داخل المتغيرات ــ الصناديق وتتبع هذه القاعدة العامة .

أحد أشكال عبارة الإسناد LET				
رقم السطر	(ثابت)	متغير	(ثابت)	رقم
	↓	↓	↓	↓
XXX	LET	XXX	=	XXX

وعلى ذلك تكون العبارة المطلوبة لتخزين الطول والعرض فى الصناديق المحددة كالتالى، مع إعطاء أرقام متتالية لهذه العبارات حسب قواعد بيسك :

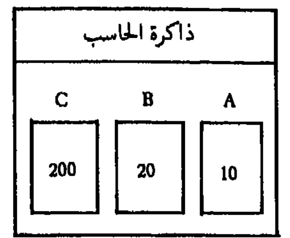
ترجمة الخطوة الأولى إلى عبارات بلغة بيسك	
١ ــ الحصول على الطول والعرض	10 LET A = 10
(البيانات المدخلة)	20 LET B = 20

الخطوة الثانية ــ(حساب المساحة ــ المعالجة) :

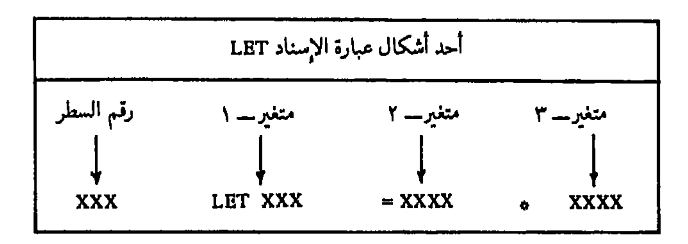
إن حساب المساحة يتطلب إسناداً من شكل آخر، فبدلا من إسناد قيمة رقمية معروفة مسبقاً، فإن القيمة التى ستخزن فى هذه الحالة ستكون نتيجة لعملية حسابية هى الضرب .

* يتم التخزين داخل الحاسب للأرقام باللغة الإنجليزية، وسنشير فيما بعد إلى طريقة لتحويل الأرقام الإنجليزية إلى عربية بهدف طباعتها .

وتتطلب هذه الخطوة تحديد اسم لمتغير جديد هو المساحة ، والوضع المطلوب هو التالى :



حيث يمثل المتغير «C» المساحة ، وكما هو الحال فى الخطوة الأولى ، تستخدم هذه الصناديق داخل عبارة الإسناد ، كما هو فى الشكل التالى :



وعليه تكون ترجمة الخطوة الثالثة كالتالى :

ترجمة الخطوة الثالثة بلغة بيسك
٢ ــ حساب المساحة (المعالجة) 30 LET C = A * B

الخطوة الثالثة ــ (إيضاح الناتج ــ المخرجات) :

إيضاح الناتج قد يتم بوسائل متعددة ، والمشهور منها والمتبع فى لغة بيسك فى كثير من الأحيان ، هو فى إيضاح الناتج على شاشة العرض (CRT) والعبارة المستعملة هى PRINT حسب الشكل التالى :

أحد أشكال عبارة الطباعة PRINT		
متغير ــ ١		رقم السطر
XXX	PRINT	XXX

وبناء على ذلك تكون ترجمة الخطوة الثالثة كما هو موضح بالشكل التالى :

ترجمة الخطوة الثالثة بلغة بيسك
٣ ــ إيضاح الناتج (المخرجات) 40 PRINT C

ويبين الشكل التالى خطوات المعالجة وترجمتها إلى عبارات بلغة بيسك :

الهدف : حساب مساحة السجاد

خطوات الحل	عبارات بيسك
١ ــ الحصول على البيانات	10 LET A = 10
الطول والعرض .	20 LET B = 20
٢ ــ حساب المساحة (المعالجة)	30 LET C = A × B
٣ ــ إيضاح الناتج (المخرجات)	40 PRINT C

ويمكنك طباعة البرنامج البسيط هذا، والانتقال إلى وضع التنفيذ Execution Mode عن طريق طبع عبارة RUN ومن ثم الضغط على مفتاح الإدخال ENTER ، وستظهر النتيجة أمامك على الشاشة .

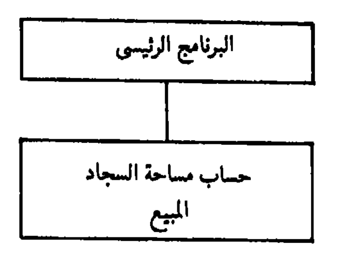
وملخص العبارات التى تم التعرض لها إلى هذه المرحلة موضحة فى الشكل التالى :

ملخص عبارات البيسك

			قيمة عددية		متغير ــ ١		
عبارة الإسناد			XXX	=	XXX	LET	XXX
	متغير ــ ٣		متغير ــ ٢		متغير ــ ١		
	XXXX	*	XXX	=	XXX	LET	XXX
					متغير		
عبارة الإخراج					XXX	PRINT	XXX

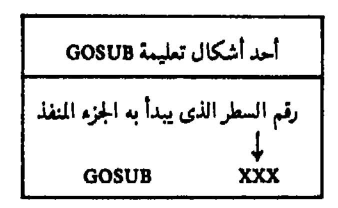
اتباع أسلوب الهيكل الهرمى فى التخطيط للبرنامج وأساليب البرمجة الهيكلية :

إن الهيكل الهرمى لهذا البرنامج يمكن تصوره وكأنه ذو جزء واحد تنفيذى، له وظيفة حساب مساحة السجاد المبيع، و يعكس الرسم التالى الهيكل الهرمى البدائى :



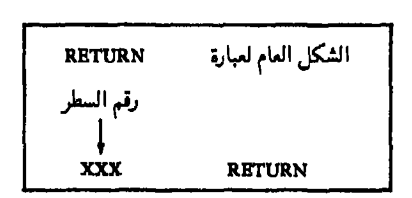
فتشكل العبارات الأربع البرنامج الخاص بالجزء، وعلينا أن نكتب البرنامج الرئيسى الذى سيتحكم فى الجزء التنفيذى . وبعمل ذلك لابد من شرح العبارة التى ستقوم بإشعار الحاسب بالذهاب إلى الجزء المنفذ، والعودة منه عقب الانتهاء إلى الجزء المتحكم .

تعليمة الذهاب إلى جزء منفذ GOSUB : تقوم هذه التعليمة بتحويل مسار تنفيذ البرنامج من الجزء المتحكم إلى الجزء التنفيذى، ولها الشكل التالى :



ولابد أن يحتوى الجزء المنفذ لتعليمة RETURN فى نهايته ؛ لإشعار الحاسب بانتهاء تنفيذ الجزء والعودة إلى الجزء المتحكم، أى إلى السطر الذى يلى السطر الذى وجدت به تعليمة GOUSB الخاصة بهذا الجزء .

والشكل العام لهذه التعليمة موضح كالتالى :



وفى حالة وجود برنامج رئيسى متحكم وآخر أو أكثر تنفيذى ، لابد من تحديد نهاية كل جزء باستخدام عبارة END ، لذلك فإن هذه العبارة تظهر بعدد الأجزاء الرئيسية والفرعية ، فى نهاية الأسطر الخاصة بكل جزء . و يكون البرنامج للمثال قيد الشرح كالتالى :

شكل (٥ ــ ١)

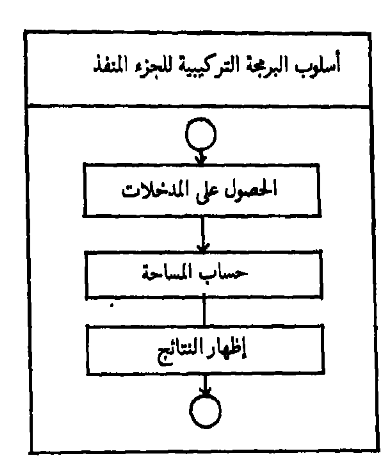
استخدام الهيكل الهرمى لكتابة برنامج
لحساب مساحة السجاد المبيع

```
10 REM                البرنامج الرئيسى المتحكم
20 GOSUB 100
30 END
100 REM               بداية البرنامج الفرعى المنفذ
110 LET A = 10
120 LET A = 20
130 LET C = A * B
140 PRINT C
150 RETURN
```

لاحظ أننا تعرضنا لتعليمة REM وهى خاصة بإضافة عبارات توضيحية تتخلل البرنامج ، وليس لها أى تأثير على سير تنفيذ البرنامج إذ أن الحاسب يتجاهلها تماما عند الوصول إليها ، و يستمر فى تنفيذ البرامج .

ولاحظ أيضا إعادة ترقيم سطور البرنامج الفرعى المنفذ فعند الوصول إلى سطر رقم 150 يعود مسار تنفيذ البرنامج إلى السطر 30 .

أما عن تصنيف أسلوب كتابة البرنامج الفرعى وفق البرمجة التركيبية ، فهو من النوع المتتالى SEQUENCE : كما هو موضح كالتالى :



أشكال أخرى للتعليمات المشروحة :

تعليمة الإسناد LET : يمكن أن تحتوى تعليمة الإسناد على متغيرات (أسماء صناديق) كثيرة بالإضافة إلى أرقام ثابتة ، كالقسمة على ٢ فى حالة حساب مساحة المثلث مثلاً .

بالإضافة إلى ذلك يمكن أن تحتوى أيضا على العمليات الحسابية كلها من ضرب وقسمة وجمع وطرح وأس . وتعكس الأمثلة التالية هذه الأشكال من استعمالات تعليمة الإسناد .

أشكال تعليمة الإسناد LET	
× × × LET X1 = X2 + X 3 / X4 × × × LET X5 = X3 / X4 – X5 * X6	
حرف X متبوع برقم اسم لمتغير (صندوق)	× × × رقم السطر

مثال : المطلوب : حساب ثمن السجاد المبيع إذا علم الطول والعرض بالأقدام ، وسعر القدم المربع ، فتكون صيغة العبارة الآن كالتالى :

```
130 LET C = A * B * P
```

حيث يمثل اسم المتغير P سعر القدم المربع . ولابد من إضافة عملية الإسناد للسعر :

```
125  LET P = 5
```

لاحظ استخدام السطر رقم 125 غير الموجود فى البرنامج السابق لإضافة التعليمة الجديدة ، فى حين تم استخدام رقم السطر القديم 130 لحساب ثمن السجاد . وعند طباعة هذين السطرين فليس هناك حاجة إلى طباعة أى من الأسطر الأخرى القديمة ، إذ أن الحاسب إما أن يضيف الأسطر الجديدة أو يستبدلها ، وذلك حسب الأرقام المعطاة لها . ويتم ذلك تلقائياً دون تدخل المبرمج وحسب تسلسل أرقام الأسطر فى البرنامج ككل .

مثال : الهدف : حساب مساحة الدائرة إذا علم نصف قطرها حسب المعادلة :

المساحة = نق^{٢} ط

حيث إن ط =٣,١٤١٥٩ ، نق = نصف القطر .

خطوات الحل	عبارات بيسك
١ ـ الحصول علي نصف القطر	110 LET N = 10
٢ ـ حساب مساحة الدائرة	120 LET M = N ↑ 2 * 3.14159
٣ ـ إيضاح النتائج على الشاشة	130 PRINT M

وفى حالة تعدد العمليات الحسابية فى عبارة الإسناد LET ، فإن تسلسل تنفيذها يخضع لقواعد لغات البرمجة كما بينت فى الفصل الرابع ، ونعيدها هنا كمراجعة .

تسلسل العمليات الحسابية فى عبارة الإسناد LET	
١ ـــ ↑	الرفع إلى قوة (أس)
٢ ـــ * /	الضرب والقسمة
٣ ـــ - +	الطرح والجمع

ففى المثال السابق يتم رفع الرقم ١٠ للأس ٢

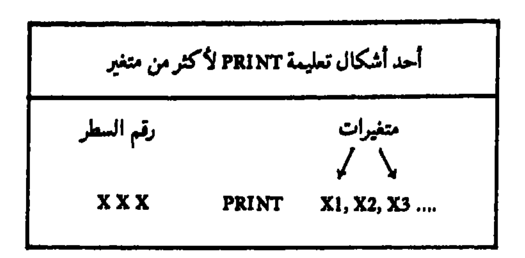
ومن ثم يضرب الناتج بالنسبة التقريبية . وفى حالة وجود عمليات حسابية متشابهة ، فإن التنفيذ لها يبدأ من بعد إشارة(«=») و يتجه إلى نهاية العبارة . ففى المثال التالى حيث إن K = 4, d = 3, C = 6, B = 5

```
1010 LET A = B * C - D * K + 3
```

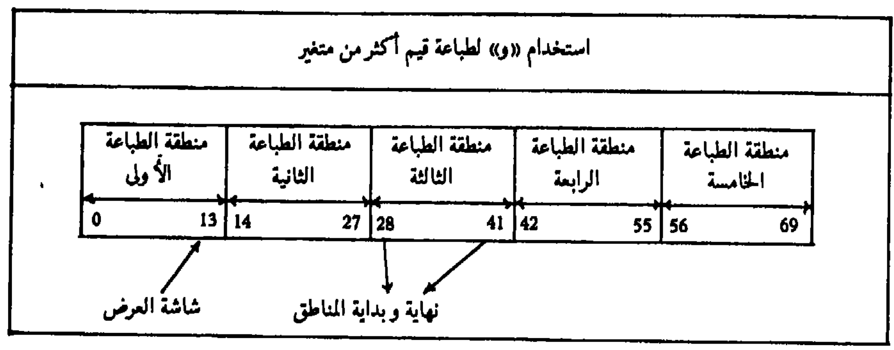
يتم التنفيذ حسب التسلسل الموضح التالى :

١ ـــ عملية الضرب الأولى	5 * 6 = 30 B * C
٢ ـــ عملية الضرب الثانية	4 * 3 = 12 D * K
٣ ـــ إشارة الطرح -	30 - 12 = 18
٤ ـــ إشارة الجمع +	18 + 3 = 21
٥ ـــ يخزن الناتج 21 فى A .	

تعليمات وأساليب إيضاح المخرجات على شاشة العرض PRINT; TAB;, : تم استخدام هذه التعليمة لإيضاح نتائج المعالجة لمتغير واحد ـــ أى محتو يات صندوق واحد ـــ ولكن قد يكون المطلوب إظهار محتو يات أكثر من صندوق ، ففى الحالة هذه يكون استخدام تعليمة PRINT كالتالى :



وتحدد قوانين لغة بيسك شكل طباعة هذه المخرجات على شاشة العرض . فالسطر على الشاشة مقسم إلى خمس مناطق طباعة PRINTING ZONES ، كل واحدة منها مؤلفة من ١٤ حرفاً* . لذلك فإن كل سطر يمكن أن يطبع فيه قيم خمسة متغيرات ، كما يوضحه الشكل التالى :

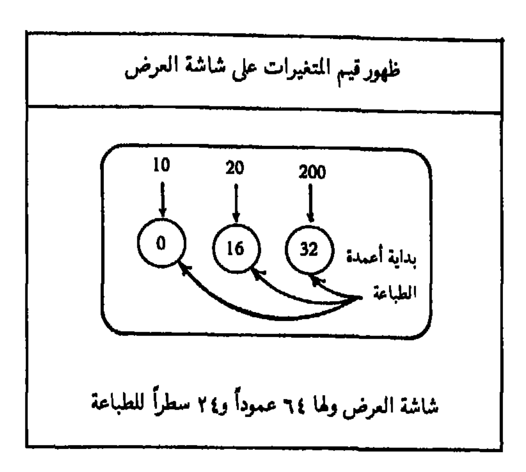


* على جهاز IBM/PC ، وتتفاوت على الأجهزة الأخرى ، ومعظمها ١٦ حرفاً مع أربع مناطق .

فلو أردنا طباعة الطول والعرض بالإضافة إلى المساحة كما فى السطر رقم 140 شكل ١ ستصبح كالتالى :

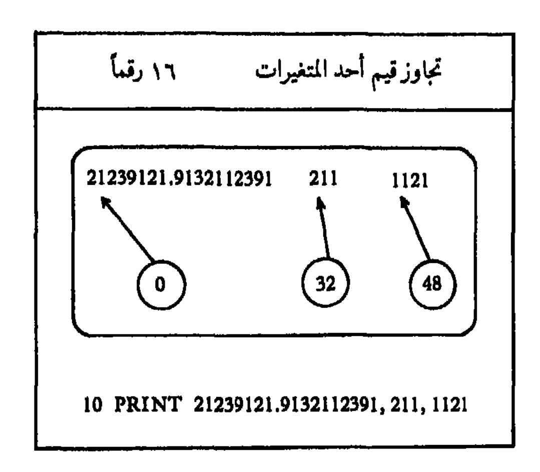
140 PRINT A, R, C

فتظهر قيم المتغيرات متسلسلة من الشمال إلى اليمين كتسلسل ظهورها فى عبارة الـ PRINT



ويخضع استخدام «و» مع تعليمة PRINT للقواعد التالية :

١ ــ إذا زاد عدد الأرقام المطبوعة لقيمة متغير واحد على ١٦ حرفاً ، فإن قيمة المتغير التالى ستطبع فى المنطقة التى تلى الحالية باثنتين . و يوضح ذلك الشكل التالى :



٢ ــ فى حال تخطى عملية الطباعة المناطق الخمس لكل سطر ، يتم الانتقال إلى السطر التالى تلقائياً . ففى العبارة التالية تطبع قيمة W فى بداية السطر التالى :

```
15 PRINT P, C, A, D, W, Z
```

استخدم الفاصلة المنقوطة; و TAB : ربما يود المبرمج ترتيب المسافات بين أرقام المتغيرات المطبوعة على الشاشة على غير ما تمليه قواعد الفاصلة«,» . فمن الممكن استخدام «;» بدلاً من «,» لطباعة الأرقام دون أية مسافات فيما بينها ــ إلا مسافة عمود واحد يتركه الحاسب بين القيم الرقمية عند إظهارها على الشاشة.

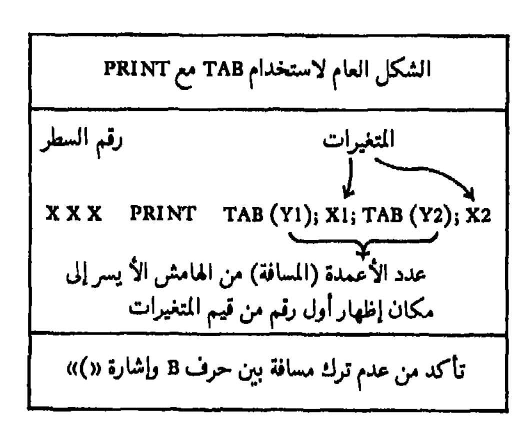
ففى نفس المثال السابق ، يظهر السطر140 على الشكل التالى :

140 PRINT A; B; C

وعليه فإن الشكل العام لاستخدام «;» يكون كالتالى :

الشكل العام لاستخدام ; مع PRINT		
المتغيرات الرقمية		رقم السطر
X1; X2; X3;	PRINT	X X X

أما TAB فتستخدم للتحكم فى المسافات بين قيم المتغيرات ــ الصناديق ــ المطبوعة . و يكون الشكل العام لها كالتالى :

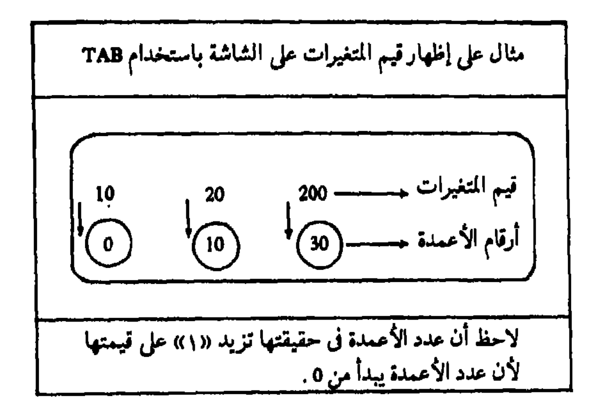


و يتم تحديد قيمتى Y1 و Y2 عن طريق معرفة عدد الأعمدة المرغوب فى تركها إلى يسار القيمة المظهرة على الشاشة ، ونلاحظ أن قيمة Y2 ، وهى لاحقة وأكبر من قيمة Y1 ، يعود تحديدها إلى العمود الصفر فى بداية الشاشة أيضاً .

ففى المثال السابق ، يمكن طباعة قيم المتغيرات C, B, A متباعدة بعضها عن بعض حسب رغبة المبرمج . ويمكن للسطر 140 أن يصبح كالتالى :

```
140  PRINT A; TAB (10); B; TAB (30); C
```

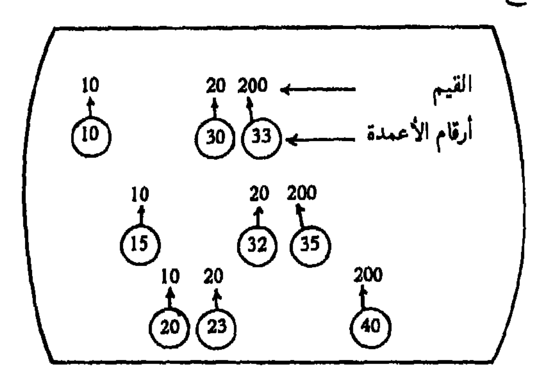
وتظهر النتيجة كالتالى :



كذلك يمكن استخدام أى من «,» و «;» و «TAB» بعضها مع بعض فى سطر واحد ، كالأمثلة التالية للسطر 140 :

```
140  PRINT TAB (10); A; TAB (30); B, C
140  PRINT TAB (15); A, B; C
140  PRINT TAB (20); A; B; TAB (40); C
```

وتظهر النتائج كالتالى :



مثال متكامل لمعالجة عملية حسابية بسيطة :

مثال (٥ ــ ٢) :

الهدف : حساب مساحة المعين إذا علم قطراه. تتضمن مراحل الحل المتكاملة الخطوات الرئيسية التالية* :

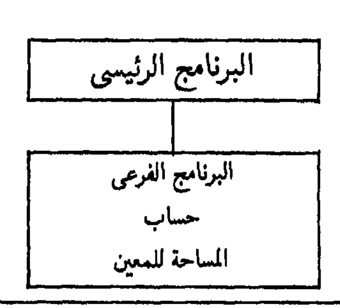
أولاً ــ خطوات الحل :

١ ــ الحصول على قيمة القطرين .

٢ ــ المساحة = القطر الأول × القطر الثانى / ٢ .

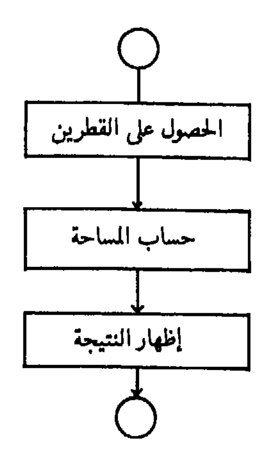
٣ ــ إظهار الناتج .

ثانياً ــ الهيكل الهرمى :



* سيتم اتباع هذه الخطوات الرئيسية الأربع فى الأمثلة القادمة .

ثالثاً ـ شكل البرمجة الهيكلية :



رابعاً ـ البرنامج :

شكل (٥ ـ ٢)

استخدام الهيكل الهرمى لكتابة برنامج لحساب مساحة المعين

```
10 REM   البرنامج الرئيسي
20 GOSUB 100
30 END
100 REM   البرنامج الفرعي لحساب مساحة المعين
110 LET A = 12
120 LET B = 8
130 LET C = A * B
140 LET D = C / 2
150 PRINT TAB(15);A;TAB(30),B
160 PRINT ,C
170 RETURN
```

ملاحظات على البرنامج :

١ ـ توزيع عمليات الإسناد على سطرين بدلاً من جمعها فى سطر واحد (110, 120) .

٢ ـ توزيع العمليات الحسابية على سطرى الإسناد 130, 140 .

٣ ــ توزيع عمليات إظهار النتائج فى سطرى الطباعة على شاشة العرض 160, 150 .
وفى سطر 160 تتم طباعة قيمة المتغير «C» ابتداء من العمود ١٦ نظراً لوجود «,»
قبلها ويمكن تكرار «,» . وكذلك إذا وضعت «,» فى نهاية السطر 150 ، فإن قيمة
C تظهر فى نفس السطر ، إذا كان هناك مساحة كافية ، وإلا فتظهر فى السطر
التالى .

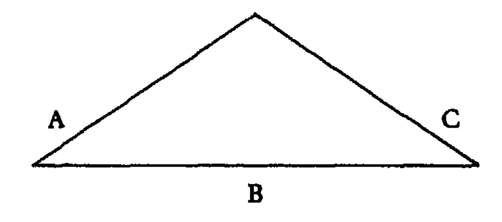
ويمكن للقارىء إعادة كتابة عبارات الطباعة على شاشة العرض باستخدام «;» و
TAB للتمارين الثلاثة السابقة .

معالجة عمليات حسابية مع أقواس :

سنقدم الآن مثالاً عن استخدام الأقواس فى العمليات الحسابية فى عملية الإسناد
إضافة إلى العمليات الحسابية الأخرى :

مثال (٥ ــ ٣) :

الهدف : إيجاد مساحة ومحيط مثلث إذا علم أطوال أضلاعه .



ملاحظة :

المحيط = مجموع أطوال الأضلاع الثلاثة

$= C + B + A$

المساحة $= \sqrt{D\,(D - C)\,(D - B)\,(D - A)}$

حيث إن $D = \dfrac{C + B + A}{2}$

أولاً ــ خطوات الحل :

١ ــ الحصول على أطوال الأضلاع عن طريق الإسناد

```
10 LET A = 5
20 LET B = 8
30 LET C = 7
```

٢ ــ حساب المحيط : `100 LET E = A + B + C`

٣ ــ إيجاد المساحة :

أ ــ إيجاد قيمة D : `200 LET D = (A + B + C) /2`

ب ــ نوجد الفرق بين كل من الأطوال الثلاثة وقيمة المتغير D .

جـ ــ نضرب ناتج الفرق بين كل من الأطوال الثلاثة بعضها ببعض ، ومن ثم نضرب الناتج فى قيمة المتغير D

```
210 LET F = (D - A) * (D - B) * (D - C)
220 LET J = F * D
```

يمكن اختصار العمليتين بعملية واحدة كالتالى :

```
210 LET F = D * (D - A) * (D - B) * (D - C)
```

ومن ثم إيجاد الجذر التربيعى ، ويمكننا عمل ذلك برفع قيمة المتغير F للقوة الثابتة 0.5 كالتالى :

```
220 LET F = F ↑ .5
```

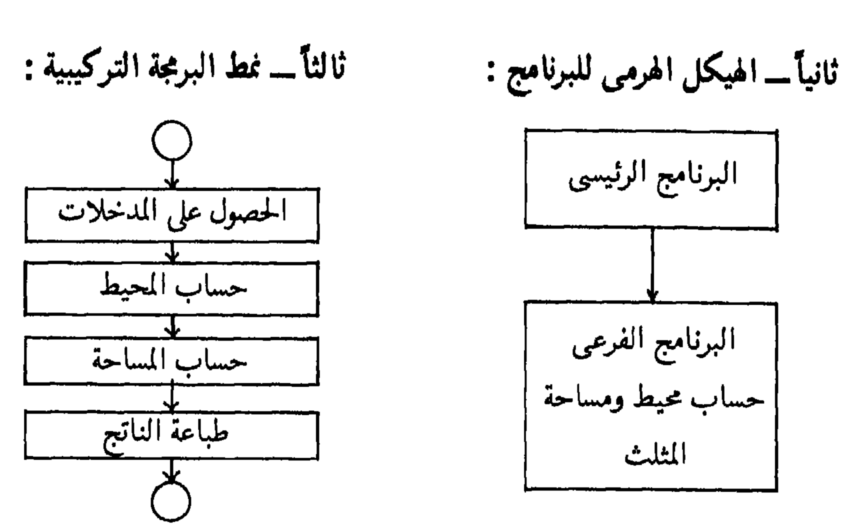
٤ ــ طباعة طول كل من الأضلاع ثم المساحة ثم المحيط .

```
250 PRINT A,B,C,F,E
```

لاحظ هنا وجود الأقواس ، أى أنه يجب حساب مجموع أطوال الأضلاع الثلاثة قبل أن نقسم على 2 . وإلا تمت قسمة القيمة التى يحملها المتغير C على 2 أولاً ومن ثم

عمليات الجمع ، وهذا يوصلنا إلى نتيجة خاطئة . أى أن معالجة العمليات الحسابية المعزولة بين الأقواس أولاً ، ومن ثم تتم العمليات الحسابية الأخرى حسب التسلسل المشروح سابقاً .

ثانياً ــ الهيكل الهرمى للبرنامج : **ثالثاً ــ نمط البرمجة التركيبية :**



رابعاً : و يتضمن الشكل التالى برنامجاً كاملاً وفق أساليب البرمجة الحديثة .

شكل (٥ ــ ٣)

برنامج لحساب محيط ومساحة أى مثلث إذا علم أطوال أضلاعه

```
10 REM   البرنامج الرئيسي
20 REM   يعمل هذا البرنامج على ايجاد مساحة و محيط مثلث
30 REM   و من ثم ايجاد المحيط و المساحه
40 GOSUB 150
50 END
150 REM   للحصول على طول كل ضلع بواسطة تعليمة اسند
160 LET A = 5
170 LET B = 8
180 LET C = 7
190 REM   للايجاد المحيط
200 LET E = A + B + C
300 REM   يجاد المساحه
310 LET D = ( A + B + C ) / 2
320 LET F = (D * ( D - A ) * ( D - B ) * ( D - C )) ^ .5
400 REM   طباعة اخراج النتائج
410 PRINT TAB(5);A;TAB(12);B;TAB(19);C;TAB(25);F;TAB(37);E
420 RETURN
```

تمارين

١ ــ اعمل على طباعة وتنفيذ البرامج الموجودة فى أشكال الفصل، وإجراء التعديلات عليها إن وجدت ، وتنفيذ البرامج حسب تسلسل هذه التعديلات .

٢ ــ بين الأخطاء إن وجدت فى القيم العددية الثابتة التالية :

521,403	254.71	$405.50	$711.53
31-415	17.705.3	875.80	220.71

٣ ــ بين الأخطاء إن وجدت فى اختيار أسماء المتغيرات العددية التالية :

A	AS	A 102	A $ 2
NAME	A + 1	B%	A.7

٤ ــ ماهى نتائج التعبيرات الحسابية التالية (مع الأخذ بعين الاعتبار بأولوية التنفيذ) إذا كانت قيمة كل من : A = 3, B = 4, C = 2

— A / B – C

— A + B * B / C + 2

— 4 + (B – C) * A – 7

— A / B / C * * 2

— (A + B) * (C – 2) * * 4

٥ ــ ماهى نتائج المعادلات الجبرية التالية إذا كانت قيم كل من :

A = 4 , B = 2 , C = 1

— $A + B - C$

— $-(A + B)$

— $(A \times B)^2 + C$

— $A^2 + AB - C^3$

— $(\frac{A}{B} + B)^2 - (\frac{C}{B}) + 4$

٦ — اعمل على تحويل المعادلات الجبرية التالية إلى تعبيرات حسابية :

— $(A \times B) + (C - B)$

— $A^2 + AB - C^{\bar{3}}$

— $\frac{(A + B)(A - B)}{(A + C)^2}$

— $A^3 - 15.5 + 7B - C$

— $(AB^2 + BC)\,B$

٧ — بين الأخطاء إن وجدت فى عبارات الإسناد التالية مع تعليل السبب :

```
10      LET      A = 70
10      LET      A + B = C
10      LET      A = «'ALT »
10      LET      A = (5 + B) - 7 + * 8
10      LET      A $ = 15 - C
10      LET      A = A + 2
10      LET      A = « 3 + 5 »
10      LET      A + B = B + A
10      LET      A + B - 7
```

٨ — بين نتائج تنفيذ العبارات التالية :

```
10      PRINT    10 , 15 , 20
10      PRINT    TAB (10) , 50 , TAB (30) , 60
```

```
10    PRINT   30 , 60 ; 70 , 100
10    PRINT   37 . 512 , 55 , 60 * 2 , 31 - 15 , 12
10    PRINT   TAB (50) , 17 , TAB (50) ; 25
10    PRINT   TAB (17) , 51 . 99 ; TAB (50) ; 91
```

٩ ــ إحدى الشركات المنفذة للمشاريع الصناعية تقوم بحساب التكلفة النهائية للمشروع كالتالى :

١ ــ تكاليف الأيدى العاملة = عدد الساعات × أجرة الساعة .

٢ ــ تكاليف الآلات = عدد الساعات × أجرة الساعة .

٣ ــ تكاليف المواد .

٤ ــ التكاليف العارضة ١٠٪ من مجموع تكاليف الأيدى العاملة والآلات .

٥ ــ التكاليف الإدارية المتعددة = ٥٪ من مجموع التكاليف السابقة كلها .

طور برنامجاً لحساب مجموع التكاليف للمشروع .

١٠ ــ تتسلم إدارة المشاريع والصيانة التابعة لوزارة الصحة طلبيات الصيانة والتى قد تتطلب القيام بواحدة أو أكثر من العمليات التالية :

الدهان ، الكهرباء ، السباكة ، النجارة ، البناء ، وبعد الانتهاء من العمليات لكل طلبية يعبأ على نفس الطلبية تكلفة كل عملية بالريالات .

اكتب برنامجاً لحساب مجموع التكلفة لكل طلبية .

الفصل السادس

برمجة معادلات بسيطة
مع عمليات إدخال

مقدمة عن عمليات الإدخال إلى الحاسب :

قدمنا فيما سبق شرحاً لأحد أساليب تخزين البيانات فى ذاكرة الحاسب ، وذلك باستخدام تعليمة الإسناد LET . ولكن هذا الأسلوب يشوبه بعض العقبات والصعاب التى تجعل استخدامه لغرض تخزين بيانات الإدخال عملية عقيمة .

ولنعد إلى أحد الأمثلة السابقة ، وهى حساب مساحة السجاد ، فكانت خطوات الحل والتعليمات المترجمة لها بلغة بيسك كما يلى (ونعيدها هنا للمراجعة والإشارة إليها) :

الهدف : حساب مساحة السجاد المبيع إذا علم الطول والعرض :

خطوات الحل	العبارات المقابلة
١ ــ الحصول على الطول والعرض	10 LET A = 10
	20 LET B = 20
٢ ــ حساب مساحة السجاد	30 LET C = A * B
٣ ــ طباعة الناتج على الشاشة	40 PRINT C

يتم طباعة البرنامج وتنفيذه للزبون الأول ، وإذا قام زبون آخر بشراء سجاد من هذا المحل ، فيجب تغيير السطرين 10 , 20 وإدخال رقمى الطول والعرض للزبون الجديد ، وهكذا مع كل بيع للسجاد . وهذا بطبيعة الحال ليس عملياً . ومحدودية هذا الاستعمال لتعليمة LET تمنع استخدامها لهذا الغرض .

إذن فلابد من وجود وسائل أخرى لإدخال البيانات إلى أسماء المتغيرات (الصناديق) دون الحاجة إلى تغيير أسطر التعليمات فى البرنامج .

وهذا الجهد الأخير يجب الاكتفاء منه بأقل القليل؛ فالبرنامج إذا حقق هدفه ، فإن تغييره دون سبب جوهرى يؤدى إلى مضيعة فى الوقت والجهد ، ويسبب الإرباك للمستفيدين من البرنامج .

وتتبع أساليب إدخال البيانات أحد الإجراءين التاليين :

١ ــ تخزين البيانات المدخلة كلها ، ومن ثم جعل البرنامج يطلبها واحدة تلو الأخرى حسب طبيعة المشكلة ، وهذا مايسمى بالمعالجة الدفعية Batch Processing .

وهذه البيانات فى برنامج مكتوب بلغة بيسك يمكن أن تخزن داخل البرنامج نفسه ، سواء فى آخره أم متناثرة خلاله ، حسب طبيعة تنفيذ البرنامج .

٢ ــ إدخال البيانات مفرزة حسب طبيعة المشكلة من لوحة المفاتيح ، ومن ثم معالجتها قبل إدخال ما تليها من البيانات ، وهذا يطلق عليه Transaction Processing .

ففى المثال السابق ، إما أن تخزن أطوال وأعراض السجاد المبيع لكل الزبائن ، ومن ثم تتم معالجتها معاً ، أو يدخل الطول والعرض لكل زبون ، ومن ثم تحسب المساحة وتطبع على الشاشة ، وبعد ذلك يتم إدخال المبيع الآخر ومعالجته بشكل إفرادى إلى حين الانتهاء منه .

وسنتعرض فى هذا الفصل إلى كيفية استخدام عبارات لغة بيسك وفق هذين الأسلوبين ولمجموعة واحدة من البيانات ، وسنكمل النقاش فى الفصل القادم لمعالجة مجموعات من البيانات .

تطوير برنامج لحل مشكلة حسابية باستخدام تعليمتى READ / DATA :

تسمح عبارة READ بتخزين البيانات المدخلة متناثرة خلال البرنامج . و يشترط أن يصاحب استخدام هذه العبارة ، طبع البيانات التابعة لها على أسطر تبدأ بعبارة DATA وذلك حتى تأخذ صبغة مميزة وفريدة مقارنة بالتعليمات الأخرى . وفيما يلى مثال على استخدام هاتين العبارتين :

مثال (٦ ــ ١) :

الهدف : حساب العمولة لمندوبى المبيعات إذا علمت النسبة ومجموع المبيعات .

أولاً ــ خطوات الحل :

١ ــ الحصول على النسبة ومجموع المبيعات .

٢ ــ حساب العمولة .

٣ ــ إظهار المخرجات على الشاشة .

إن الفرق بين هذه الخطوات والتى سبقتها هو فى ترجمة الخطوة الأولى . فبدلاً من استخدام أسلوب الإسناد سنستخدم أسلوب إدراج البيانات داخل البرنامج . والشكل العام لاستخدام عبارة READ كما يلى :

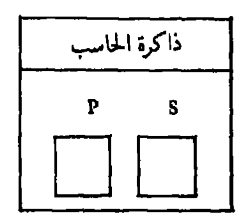
الشكل العام لعبارة READ
XXX READ X 1, X 2, X 3
رقم السطر ← XXX ؛ أسماء المتغيرات ← X 1, X 2, X 3

و يلاحظ أن عدد أسماء المتغيرات التى تظهر بعد كلمة READ يتأثر بعدد البيانات المدخلة . وفى مثالنا هذا ، نحتاج إلى صندوقين داخل الذاكرة : واحد للعمولة ، وآخر للمبيعات ، وعليه فإن العبارة تكون فى شكلها النهائى :

10 READ P , S

حيث يرمز P إلى العمولة و S للمبيعات بالريال .

أما فى الذاكرة ، فتكون هذه المتغيرات كالتالى :



وفى حالة تنفيذ هذه العبارة ، يبحث البرنامج عن أول سطر تظهر فيه عبارة DATA ، وتخزن اثنتان من البيانات يفصل بينهما «,» .

ثانياً وثالثاً ــ يشبه الهيكل الهرمى ونمط البرمجة التركيبية : المثال الأخير فى الفصل السابق. **رابعاً** ــ ويمثل الشكل التالى البرنامج كاملاً وفق الأساليب الحديثة فى البرمجة.

شكل (٦ ــ ١)

برنامج لحساب العمولة لمندوبى المبيعات

```
10 REM  البرنامج الرئيسي
20 GOSUB 100
30 END
100 REM  برنامج فرعي لحساب العمولة
110 REM  لقراءة العمولة والمبيعات
120 READ P,S
130 LET M = P * S
140 REM لطباعة نسبة العمولة، المبيعات و العمولة
150 PRINT P,S,M
160 DATA .10,10000
170 RETURN
```

ملاحظات على البرنامج :

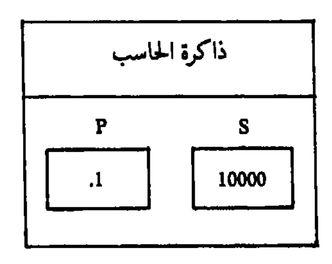
١ ــ تكون محتويات المتغيرات (الصناديق) على الشكل التالى :

ذاكرة الحاسب الآلى		
P	S	M
.10	10000	100

٢ ــ من الممكن أن تظهر عبارتا READ / DATA على النحو التالى :

أشكال من اقتران READ مع DATA	
١ 105 READ P 110 READ S 140 DATA .1,10000	٤ 110 READ P , S . . . 140 DATA .1 150 DATA 10000
٢ (مع إلغاء 140) 105 READ P 107 DATA .1 110 READ S 112 DATA 10000	٥ (مع إلغاء 140) 105 DATA .1,000 110 READ P, S
٣ 105 READ P 110 READ S . . . . 140 DATA .1 150 DATA 10000	٦ 105 READ P, S 140 DATA .1, 10000

وفى كـل هـذه الأشكال تكون نتيجة تنفيذ هذه العبارات تخزين القيم 1. ، 10000 فى ذاكرة الحاسب على النحو التالى :



وذلك لأن تخزين قيمتى المتغيرين S, P يتم عن طريق ترتيب ظهور هذين المتغيرين فى تـعـلـيـمـة READ ، وظـهـور قـيـمتيهما المواكبتين لهما فى تعليمة DATA . لاحظ فى الأسـلـوب رقـم ٦ أن ظهور « ، » فى نهاية سطر DATA تعنى استمرار البيانات فى السطر الذى يليه دون الحاجة إلى طباعة رقم السطر وكلمة DATA .

فـالمبرمج عليه الحرص دائماً على موافقة ترتيب أسماء المتغيرات ــ الصناديق ــ مع قيمها المطابقة لها ، وإلا فإن الحاسب لن يكتشف الخطأ فى التسلسل .

وسـنـقـدم الآن مـثالاً آخر لاستخدام عبارتى READ / DATA ، وإعطاء أمثلة عن بعض الأخطاء التى يقع فيها المبتدئون فى أكثر الأحيان .

مثال (٦ ــ ٢) :

الهدف : إيجـاد المـعـدل الـعام لعلامات دارس تقدم إلى امتحان قبول مكون من خمسة أجزاء .

أولاً ــ خطوات الحل :

١ ــ قراءة علامات الأجزاء الخمسة .

٢ ــ إيجاد المجموع .

٣ ــ إيجاد المعدل العام .

٤ ــ طباعة العلامات الخمس والمعدل العام .

ثانياً وثالثاً ــ كما فى المثال السابق .

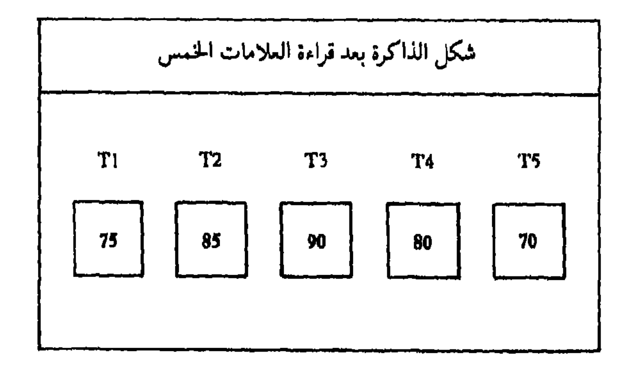
رابعاً ــ الحل كاملاً لهذا المثال حسب الأساليب الحديثة فى البرمجة .

شكل (٦ ــ ٢)

برنامج لحساب معدل خمس علامات

```
10 REM  استدعاء برنامج فرعي لحساب المعدل العام و طباعة النتائج
20 GOSUB 100
30 END
100 REM  لقراءة الدرجات الخمسة و ايجاد المعدل العام و طباعته
110 READ T1,T2,T3,T4,T5
120 DATA 75,85,90,80,70
130 LET A = ( T1 + T2 + T3 + T4 + T5 ) / 5
140 PRINT TAB(5);T1;TAB(10);T2;TAB(15);T3;TAB(20);T4;TAB(25);T5;TAB(30);A
150 RETURN
```

وستظهر ذاكرة الحاسب بعد تنفيذ هذا البرنامج كالتالى :



ومن الأخطاء الممكن حدوثها هى :

١ ـــ نقصان عدد القيم فى عبارة DATA عن عدد أسماء المتغيرات فى عبارة READ . ففى المثال السابق لنفرض أن المبرمج نسى أن يطبع القيمة الأخيرة (70) فى عبارة DATA فى السطر 80 ، بحيث ظهر سطر 80 كالتالى :

```
80 DATA 75 , 85 , 90 , 80
```

ففى هذه الحالة يتوقف تنفيذ البرنامج مشيراً إلى وجود نقص فى البيانات .

٢ ـــ زيادة عدد قيم المتغيرات عن عدد أسماء المتغيرات ، وهذا عكس الحالة السابقة . ففى هذه الحالة يتجاهل الحاسب القيم الزائدة و يستمر فى تنفيذ البرنامج . فإذا كانت القيمة الزائدة لا تؤدى إلى تغيير فى طبيعة القيم فلن يؤثر ذلك على النتيجة .

فإذا ظهر السطر 80 كالتالى :

```
80 DATA 75 , 85 , 90 , 80 , 70 , 85
```

فستكون نتيجة حساب المعدل صحيحة .

أما إذا كان الخطأ فى سطر 80 كالتالى :

```
80 DATA 75 , 8,5 , 90 , 80 , 70
```

أى أن المبرمج طبع القيمة 85 كقيمتين 8 , 5 . فهذا يؤدى إلى خطأ فى نتيجة حساب المعدل ، على الرغم من أن الحاسب لم يكتشف وجوده .

٣ ـــ عدم توافق بين تسلسل أسماء المتغيرات (صناديق) والقيم المراد تخزينها فيها .

ففى المثال السابق إذا تم إبدال تسلسل أى علامتين كما فى السطر التالى :

```
80 DATA 85 , 75 , 90 , 80 , 70
```

فلن يؤثر هذا الخطأ على العملية الحسابية ، ولكن فى بعض الأحيان يؤدى هذا التبديل إلى خطأ فى حساب المعدل .

فإذا تم تعديل طريقة حساب المعدل بحيث أعطيت نسب مئوية مختلفة لكل واحدة من العلامات ، أصبحت عبارة 90 كالتالى :

```
90  LET A = T1 * .2 + T2 * .15 + T3 * .25 + T4 * .10+T5 * .30
```

أى أن نسب العلامات الخمس بالتوالى كانت :

20 % , 15 % , 25 % , 10 % , 30 %

ففى الخطأ المبين سابقاً فى عبارة 80 ستكون نتيجة حساب المعدل خاطئة ؛ بسبب تبادل نسب العلامة الأولى والثانية .

فيما سبق من هذا الفصل ، تحدثنا عن قراءة ومعالجة بيانات رقمية حسابية ، ولكن كثيراً من المشاكل التى تتم معالجتها بواسطة الحاسب تتضمن بيانات غير رقمية ولا تستعمل للعمليات الحاسبية ، كاسم الموظف وعنوانه ودائرته التى يعمل بها واسم وظيفته .

وسنتعرض الآن لكيفية التعامل مع البيانات غير الرقمية .

تطوير برنامج باستخدام بيانات حرفية (غير رقمية) وتعريف المخرجات :

ففى المثال السابق ، لنفرض أننا أردنا تعريف اسم الدارس الذى تخصه هذه العلامات ، وهذا يتطلب منا استخدام اسم متغير غير حسابى لتخزين اسم الدارس به .

وفى اختيار اسم المتغير غير الحسابى تنطبق نفس الشروط التى تنطبق على اختيار اسم المتغير الحسابى (ارجع إلى الفصل الخامس) مع إضافة إشارة الدولار $ فى نهاية اسم المتغير .

لذا فستعدل العبارات التالية :

```
70   READ  A $ , T1 , T2 , T3 , T4 , T5
80   DATA « AHMED ALI » , 75 , 85 , 90 , 80 , 70
```

لاحظ هنا أن البيانات غير الحسابية يجب وضعها بين علامتى التنصيص فى عبارة DATA للدلالة على نوعيتها غير الحسابية . و يتغير السطر 100 ليصبح كالتالى :

```
100 PRINT TAB (5); A$; TAB (15); T1; TAB (22); T2; TAB (29);  T3;
    TAB (36); T4; TAB (43); T5; TAB (50); A
```

لاحظ أن المتغير A$ تمت إضافته فى العبارة 70 فى بداية المتغيرات وثم وضع «AHMED ALI» فى بداية البيانات (80) أى أن البيانات قد رتبت على حسب ما يرادفها من أسماء المتغيرات ونوعيتها .

ولو قمنا بتشغيل البرنامج شكل (٦ ــ ٢ أ) بعد إتمام التعديلات عليها ، لوجدنا أن المخرجات ستكون على الشكل التالى :

```
Ahmed  Ali  75  85  90  80  70
```

دون أى تمييز لما تمثله أرقام العلامات .

فإذا أردنا تحديد الامتحانات والعلامة التى حصل عليها Ahmed Ali فى كل منها ، يمكننا عمل ذلك باستخدام تعليمة اطبع RPINT ، وهذا النوع من العمليات يطلق عليه «عمليات التعريف لحقول البيانات المخرجة عن طريق وضع عناو ين لكل منها» ، و يتم ذلك قبل البدء فى طباعة أى من مخرجات البرنامج .

ففى مثالنا السابق إذا أردنا وضع عناو ين لكل من حقول البيانات نستطيع عمل ذلك بإضافة العبارة التالية :

15 PRINT TAB (5); «NAME»; TAB (5); «TEST1»; TAB (20); «TEST2»; TAB (25); «TEST3»; TAB (30); «TEST4»; TAB (35); «TEST5»; TAB (40); «AVERAGE»

ولفصل العناو ين عن البيانات المخرجة لتسهيل عملية قراءة وملاحظة المخرجات ، نعمل على وضع خط فاصل بإدخال العبارة التالية :

17 PRINT TAB (5); «..........»; «TAB (15); «..........»; TAB (20); «..........»; TAB (25); «..........»; TAB (30); «..........»; TAB (35); «..........»; TAB (40); «..........»;

ولنر الحل كاملاً للمثال شكل (٦ ــ ٢) بعد إجراء التعديلات عليه :

شكل (٦ ــ ٢ أ)

برنامج لحساب معدل خمس علامات لدارس واحد مع طباعة الاسم والعناو ين

```
10 REM  استدعاء برنامج فرعي لحساب المعدل العام و طباعة النتائج
20 GOSUB 40
30 END
40 REM  لطباعة العناوين
50 PRINT TAB(5);"NAME      ";TAB(15);"TEST-1";TAB(24);"TEST-2";TAB(32);"TEST-3";T
AB(40);"TEST-4";TAB(49);"TEST-5";TAB(59);"AVERAGE"
60 PRINT TAB(5);"---------";TAB(15);"------";TAB(24);"------";TAB(32);"------";TA
B(40);"------";TAB(49);"------";TAB(59);"-------"
70 REM  للقراءة الدرجات الخمسة و ايجاد المعدل العام و طباعته
80 READ N$,T1,T2,T3,T4,T5
90 DATA "AHMED ALI",75,85,90,80,70
100 LET A = ( T1 + T2 + T3 + T4 + T5 ) / 5
110 PRINT TAB(5);N$;TAB(15);T1;TAB(24);T2;TAB(32);T3;TAB(40);T4;TAB(49);T5;TAB(5
9);A
120 RETURN
```

ولو قمنا بتشغيل البرنامج شكل (٦ ــ ٢ أ) لوجدنا أن المخرجات ستكون على الشكل التالى :

```
NAME       TEST-1    TEST-2   TEST-3   TEST-4    TEST-5     AVERAGE
---------  ------    ------   ------   ------    ------     -------
AHMED ALI   75        85       90       80        70          80
```

تكلمنا فى الجزء السابق من هذا الفصل عن كيفية تخزين البيانات فى البرنامج المصدرى عن طريق استخدام تعليمتى READ / DATA . وسنتحدث الآن عن كيفية إدخال البيانات بواسطة لوحة المفاتيح وذلك باستخدام تعليمة أدخل INPUT .

تطوير برنامج باستخدام تعليمة أدخل INPUT :

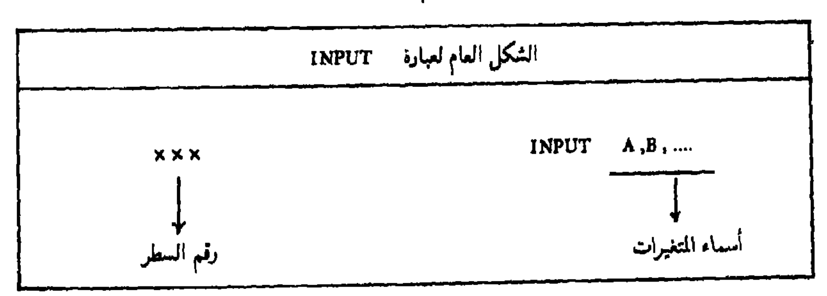
مثال (٦ ــ ٣) :

الهدف : إيجاد مساحة ومحيط مستطيل إذا علم طوله وعرضه .

أولاً ــ خطوات الحل :

١ ــ الحصول على طول وعرض المستطيل .
٢ ــ إيجاد المساحة .
٣ ــ إيجاد المحيط .
٤ ــ طباعة الطول ، العرض ، المساحة ، والمحيط .

وإليك الشكل العام لعبارة INPUT

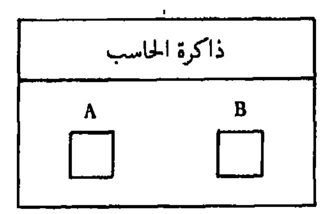


إن عدد قيم البيانات المراد إدخالها يتأثر بعدد أسماء المتغيرات التى تظهر بعد كلمة INPUT .

ففى مثالنا هذا نحتاج إلى صندوقين داخل الذاكرة : واحد للطول ، وآخر للعرض .

وعليه فإن العبارة ستكون : 10 INPUT A , B

حيث يرمز A للطول و B للعرض ، أما فى الذاكرة فتكون هذه المتغيرات كالتالى :



وفى حالة تنفيذ هذه العبارة سيتوقف تنفيذ البرنامج منتظراً من المستخدم إدخال قيمتين ، ومن ثم إشعاره بالانتهاء من ذلك عن طريق ضغط مفتاح ENTFR بعد طباعة البيانات .

ثانياً وثالثاً ــ كما فى المثال السابق.

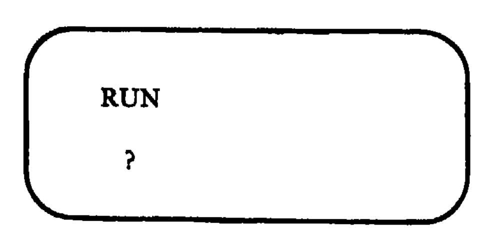
رابعاً : ويمثل الشكل التالى برنامجاً كاملاً وفق الأساليب الحديثة فى البرمجة .

شكل (٦ ــ ٣)

برنامج لحساب مساحة ومحيط مستطيل إذا علم طوله وعرضه

```
10 REM  استدعاء برنامج فرعي لحساب المساحه والمحيط
20 GOSUB 100
39 END
100 REM  برنامج فرعي للحصول على طول و عرض المستطيل
110 INPUT A,B
120 REM  لحساب المساحه
130 LET C = A * B
140 REM  لحساب المحيط
150 LET D = ( A + B ) * 2
160 REM  لطباعة الطول ,العرض , المساحه و المحيط
170 PRINT A,B,C,D
180 RETURN
```

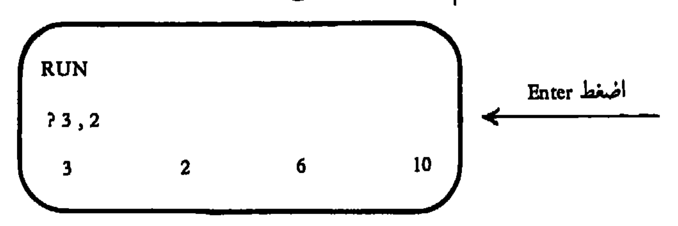
فلو قمنا بتشغيل هذا البرنامج فستظهر علامة استفهام على شاشة النهاية الطرفية ، مشيرة إلى أن تنفيذ البرنامج قد توقف عندما وصل إلى عبارة 110 منتظراً من المستخدم أن يدخل قيمتى كل من الطول والعرض B , A كما هومبين فيما يلى ، ليعمل على تخزينهما (القيمتين) فى الذاكرة الرئيسية .



وكما هو الحال فى تعليمة DATA بأن يجب أن تكون البيانات مفصولة بعضها عن بعض بفاصلة .

فهنا يقوم المستخدم بواسطة لوحة المفاتيح بإدخال القيمتين مفصولتين بفاصلة . فلو تم إدخال 2 , 3 سيبقى التنفيذ متوقفاً ، إلى أن يتم إشعار البرنامج بالانتهاء من إدخال القيم بالضغط على مفتاح العودة RETURN أو ENTER حسب نوع الجهاز المستخدم (وفى كليهما ستكون الوظيفة واحدة) . و بعد الإشعار ، سيستمر تنفيذ البرنامج .

فى مثالنا هنا سيتم التنفيذ كما يلى :



المذكرات التوضيحية : لاحظ أن علامة الاستفهام ظهرت على الشاشة ، دون أن يسبقها أى تفسير لنوعية وترتيب وعدد البيانات . هذا يؤدى إلى غموض و بلبلة لمستخدم البرنامج .

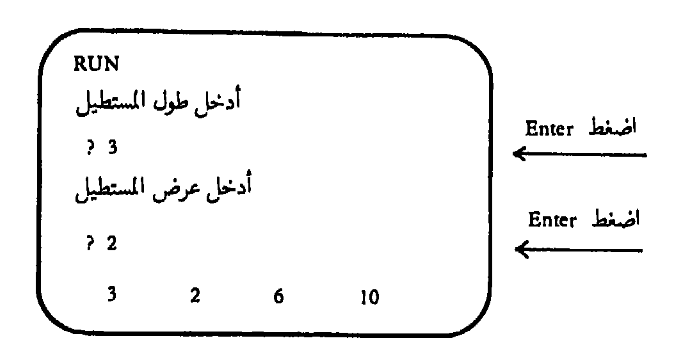
لذا فمن المستحسن أن يسبق عبارة INPUT طباعة مذكرة توضيحية كوسيلة لتعريف حقول البيانات ، وتسهيلاً للمستخدم لتحديد أى منها يراد إدخاله ، وتسلسلها المطلوب . و يتم ذلك باتباع أحد أسلوبين :

أ ــ باستخدام تعليمة اطبع PRINT فى سطر يسبق السطر الذى تظهر معه تعليمة أدخل INPUT . و يتم تعديل البرنامج السابق كالتالى :

```
100 PRINT        «أدخل طول المستطيل»
112 INPUT A
114 PRINT        «أدخل عرض المستطيل»
116 INUT B
```

لاحظ هنا أن المفتاح ENTFR أو RETURN سيضغط مرتين ، الأولى : بعد طباعة الطول 3 ، والثانية : بعد طباعة العرض 2 .

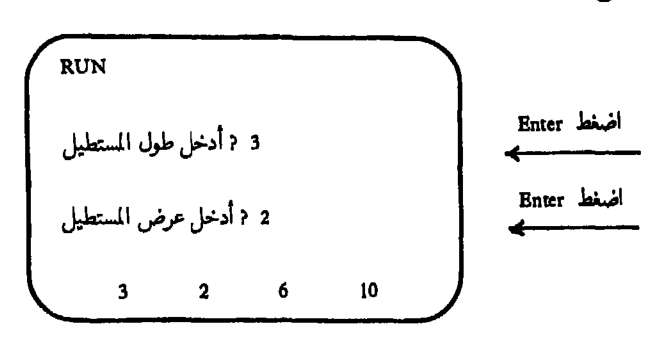
إذا تم تشغيل البرنامج فسيظهر التالى :



ب ــ بطباعة المذكرة التوضيحية فى السطر الذى تظهر فيه تعليمة INPUT و بعدها مباشرة ، وذلك كالتالى :

```
110 INPUT   «أدخل طول المستطيل» ; A
116 INPUT   «أدخل عرض المستطيل» ; B
```

لاحظ رمز «;» الذى يسبق أول متغير يظهر فى عبارة INPUT . وإذا تم تشغيل البرنامج فسيظهر كالآتى (وذلك بعد محو السطرين 112 , 114)



وعند تشغيل البرنامج سيتبين المغزى من استخدام تعليمة أدخل INPUT مع المذكرة التوضيحية :

١ ـ اتباع نظام المخاطبة (التعامل بطريقة مباشرة) مع الحاسب .

٢ ـ إمكانية استخدام بيانات مختلفة فى كل مرة يتم فيها تنفيذ البرنامج دون الحاجة للتغيير فى البرنامج المصدرى (تغيير البيانات الموجودة فى عبارة DATA) .

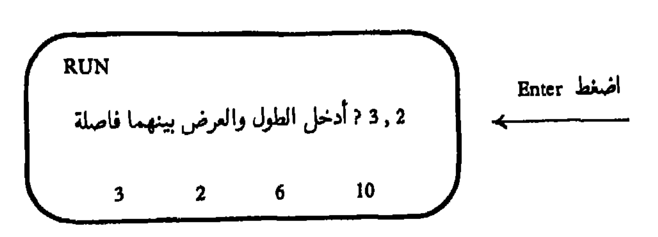
٣ ـ عدم المزج فى نوعية وترتيب (تسلسل) البيانات .

٤ ـ إدخال البيانات المراد التعامل معها دون زيادة أو نقصان .

وكذلك من الممكن كتابة البرنامج بأسلوب تطلب فيه القيمتان فى سطر واحد ، وذلك كالتالى :

110 INPUT «أدخل الطول والعرض بينهما فاصلة» ; A , B

وعند تشغيل البرنامج سيحدث التالى (بعد محو سطر 116) :



تمارين

١ ــ اعمل على طباعة وتنفيذ البرامج الموجودة فى أشكال الفصل ، وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعديلات .

٢ ــ بين الأخطاء إن وجدت فى كل من العبارات التالية :

```
10    READ     A , B , C
10    INPUT    A , B , C
10    READ     6 A , A £ , C
10    INPUT    AB ; C ;
10    DATA     100 , 200 AB
10    DATA     500 , 600 , 101 ;
```

٣ ــ بين الأخطاء إن وجدت فى أجزاء البرنامج التالية :

```
10    READ     A , B , C
20    LET      A = A + BC
30    DATA     10 , 20
10    INPUT    A , B
20    LET      C = A * B
30    PRINT    A , B A * B
```

٤ ــ بين نتائج تنفيذ البرامج التالية :

```
10    READ     A , B , C
20    LET      D = A + B * C
30    PRINT    A , B , C , D
40    DATA     10 , 5 , 2
50    END
```

```
10    READ      N$ T1 , T2
20    DATA      « ALI  AHMED »  , 70 , 90
30    LET       A = (T1 + T2) / 2
40    PRINT     TAB 10 , N$ , T1 , T2 , A
50    END
```

٥ ــ باستخدام تعليمة أدخل INPUT اكتب برنامجاً يعمل على استقبال اسم الدارس وعنوانه المؤلف من : صندوق البريد ، المدينة ، والرمز البريدى ، ومن ثم طباعة هذه البيانات حسب الصيغة التالية :

الاسم :

ص . ب :

اسم المدينة :   الرمز البريدى :

ملاحظة : استخدم تعليمات الإيضاح لتحديد البيانات المطلوب إدخالها .

٦ ــ يقوم دهّان (طراش) منازل بحساب تكلفة دهان المنزل على الأساس التالى :

التكلفة الكلية = (المواد + الأيدى العاملة) × عامل الجودة .

ولتقدير التكلفة الكلية يقوم الدهان بحساب مساحة الجدران بالأمتار المربعة ويطرح منها مساحة الشبابيك . وتكلفة المواد هى ٧٠ ريالاً لكل ٥ أمتار مربعة . أما الأيدى العاملة فتكلف ١٢٠ ريالاً لكل ١٠ أمتار مربعة . أما عامل الجودة فيتراوح بين ٩٠٪ ــ ١١٠٪ وذلك حسب طبيعة الدهان فى المنزل الحالى ، وطبيعة الجدران . فإذا كانت الجدران فى وضع مقبول ولا تحتاج من التجهيز إلا إلى الجهد العادى ، فالنسبة تكون ١٠٠٪ . أما الجدران التى تتطلب تجريدها من الدهان القديم وفيها خروق ... إلخ ، فتكون النسبة ١١٠٪ ، أما الجدران التى فى وضع جيد ولا تتطلب أى تجهيز (كالمنازل الجديدة) فالنسبة هى ٩٠٪ ... وهكذا .

طور برنامجاً لحساب تكلفة المنازل إذا علمت المدخلات التالية :

مساحة الجدران ، مساحة الشبابيك ، عامل الجودة .

الفصل السابع

الدوارة البسيطة والتحكم بها

مقدمة عن استعمالات الدوارة فى معالجة البيانات :

فى الفصلين السابقين تحدثنا عن برمجة خطوات متتالية لحل مشكلة حسابية بسيطة . وانتقلنا فى العرض من إسناد قيم المدخلات إلى متغيراتها (الصناديق) ، باستخدام تعليمة LET ، ومن ثم باستعمال تعليمتى READ/ DATE ، وأخيراً باستعمال تعليمة INPUT . وذلك لقراءة البيانات الموجودة داخل البرنامج المصدرى ، أو المدخلة عن طريق لوحة المفاتيح على التوالى .

وقد أوضحنا كيف أن الانتقال فى عملية قراءة البيانات قد أدى إلى زيادة وضوح وفاعلية البرنامج . فمع عملية الإسناد ، استدعى تغيير قيمة المتغيرات فى البرنامج نفسه . أما READ / DATE فالتغيير يتم فى عبارة DATA و يتم استخدام نفس المتغيرات فى عبارة READ . وأخيراً فى تعليمة INPUT يتم تنفيذ البرامج بطباعة RUN مرات عديدة حسب تكرار البيانات المدخلة .

وهذا التبسيط فى أسلوب البرمجة يهدف إلى التدرج فى عملية البرمجة ، إذ أن واقع الحال يتطلب عبارات متعددة أخرى ومعقدة و يصبح معها هذا الأسلوب عقيماً أيضاً . و يوضح المثال التالى عدم فاعلية هذا الأسلوب ، وكيفية تطو يره ليتلاءم مع متطلبات التطبيقات المختلفة ، وسنستخدم نفس المثال طوال هذا الفصل لشرح التعليمات الخاصة بالتحكم بالدوارة البسيطة .

مثال (٧ ــ ١) :

الهدف : حساب محيط الدائرة إذا علم نصف قطرها .

خطوات الحل ومايقابلها من عبارات بيسك :

١ ــ الحصول على المدخلات 10 READ N

20 DATA 10

٢ ــ حساب المحيط = ع نق ط 30 LET M = 4 * N * 3.147

حيث نق = نصف القطر

ط =٣,١٤٧

٣ ــ طباعة النتائج 40 PRINT N,M

وإذا أردنا تنفيذ هذا البرنامج لحساب محيط دائرة أخرى ذات قطر مختلف ، فعلينا تغيير سطر رقم 20 وإعادة طباعة القيمة الجديدة لنصف القطر . وهذا ليس عملياً تماماً ، وخاصة إذا أردنا حساب المحيط لدوائر كثيرة . لاحظ هنا أن الخطوات لإيجاد محيط الدائرة لا تتغير ، وإنما تبقى كما هى ، فقط تتغير قيمة نصف القطر الخاص بكل دائرة .

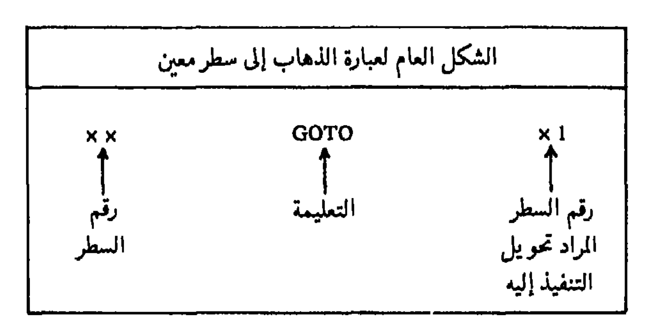
وتحدثنا فى الفصل السابق عن إمكانية طباعة قيم متعددة فى عبارة DATA . والفرق هنا أن هذه القيم كانت تعود إلى عنصر واحد ــ الطالب مثلا . ولكن القيم فى مثالنا هنا تعود إلى عناصر موصوفة ــ الدائرة هنا ــ مختلفة ، ولكن المبدأ واحد . أى أنه باستطاعتنا أن ندرج فى السطر 20 جميع قيم أنصاف أقطار الدوائر المراد حساب محيطاتها . فتظهر العبارة كالتالى :

20 DATA 10 , 15 , 20 , 25 , 5 , 3 , 12 , 15 , 12

أى أن هناك ٩ دوائر يراد حساب محيطاتها ، بالإضافة إلى هذه الخاصية ، فلابد من وجود عبارة توجه تنفيذ البرنامج إلى الخطوة الأولى مرة أخرى أو مرات ، طبقاً لعدد القيم الموجودة فى عبارة DATA . وهذا ما سنبحثه فى الجزء التالى .

تطوير برنامج بواسطة استخدام تعليمة الذهاب إلى رقم سطر معين :

إن تعليمة الذهاب إلى رقم سطر معين تفرض مساراً محدداً لاتجاه تنفيذ البرنامج دون أية شروط مسبقة ، لذلك يطلق عليها تعليمة التشعب غير المشروط للتحكم فى مسار تنفيذ خطوات البرنامج . أى أن الحاسب عند تنفيذه هذه التعليمة ، سيحول مسار التنفيذ إلى السطر المذكور دون الأخذ بعين الاعتبار بأية شروط أو لأى من قيم المتغيرات الموجودة فى البرنامج . و يوضح الشكل التالى الصورة العامة لهذه التعليمة .



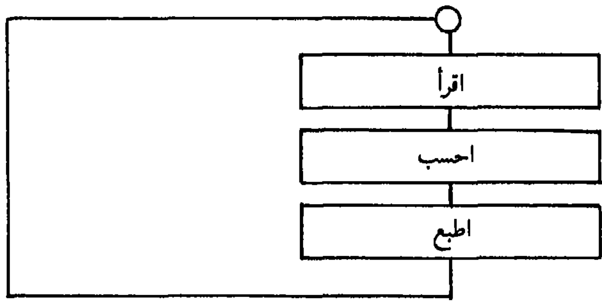
وفيما يلى خطوات تطوير برنامج للمثال السابق (٧ ــ ١) :

أولاً ــ خطوات الحل :

إضافة خطوة رقم ٤ ــ اذهب إلى الخطوة الأولى.

ثانياً ــ الهيكل الهرمى : كما فى البرامج السابقة ، إلى برنامج جزئى واحد .

ثالثاً ــ نمط البرمجة الهيكلية :



رابعا ــ البرنامج الخاص بحساب محيط الدائرة :

شكل (٧ ــ ١)

برنامج لحساب محيط الدائرة باستخدام عبارة GOTO

```
10 REM   البرنامج الرئيسي
20 REM   استدعاء برنامج فرعي
30 GOSUB 100
40 END
100 REM   برنامج فرعي لحساب محيط الدائره
110 READ N
120 DATA 10,15,20,25,5,3,12,45,12
130 LET M = 4 * N * 3.14
140 PRINT N;" نصف القطر "
150 PRINT M;" محيط الدائره "
160 GOTO 110
170 RETURN
```

إن البرنامج السابق يبدو لأول وهلة أنه برنامج صحيح ، ولكن إذا أمعنت فى النظر فيه ، فستجد أن إضافة عبارة الذهاب إلى السطر 160 أدى إلى تكرار تنفيذ سطور البرنامج دون أى إشارة أو تبيان لأسلوب توقيفها . فكما قلنا سابقا ، إن عبارة GOTO تدعو إلى التشعب غير المشروط .

فإذا تم تنفيذ البرنامج ، فإنه فى المرة الأولى التى تنفذ فيها السطور ، يتم أخذ قيمة 10 وهى أول قيمة فى عبارة DATA ، وتخزن فى المتغير N ، وفى المرة الثانية 15 ، وهكذا إلى نهاية القيم ، ماسحة القيم السابقة التى خزنت فى N فى كل مرة حين قراءة القيمة

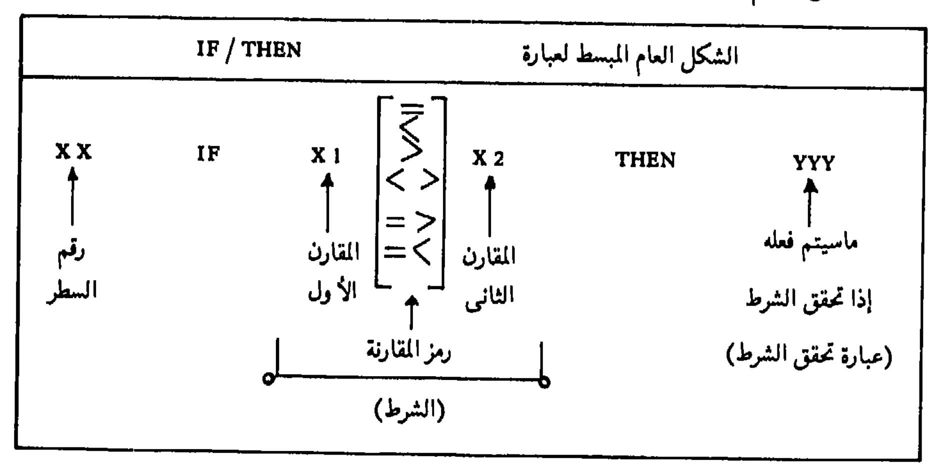
الجديدة . وبعد الانتهاء من القيمة الأخيرة ، 12 فإن البرنامج يعود إلى تنفيذ السطور مرة أخرى ، و يذهب باحثاً عن قيمة جديدة للمتغيرN حين تغرضه للسطر110 ، ولكن هذه القيم تكون قد نفدت . وحينئذ يتوقف البرنامج عن التنفيذ ، مشيراً إلى وجود خطأ ناتج عن عدم احتواء عبارة DATA على بيانات أخرى .

وبعبارة أخرى ، فإن البرنامج لا يستطيع معرفة ما إذا نفدت قيم المتغيرات تلقائياً ، وعلى المبرمج أن يتحكم بذلك عن طريق خطوات البرنامج .

وهذا يدعونا إلى التحدث عن أساليب التحكم فى قراءة البيانات ؛ وذلك منعاً لتوقف البرنامج بصورة غير طبيعية عند نفادها . و يتم ذلك عن طريق استخدام تعليمة التشعب المشروط IF / THEN .

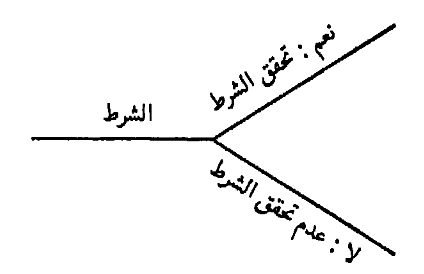
كيفية عمل تعليمة إذا / إذن IF / THEN بشكلها المبسط :

تستخدم هذه التعليمة للتحكم فى مسار تنفيذ أسطر البرنامج بناء على شرط محدد ، ولها الشكل العام التالى :



ولعبارة IF/THEN جزءان رئيسيان :

١ ــ الــشــرط : إن جزء الشرط هو الذى سيحدد اتجاه مسار التنفيذ . وتكون نتيجة الشرط مرهونة بعملية المقارنة بين عنصرين من البيانات ، أطلق عليهما «المقارن الأول» و «المقارن الثانى» . وتكون هذه النتيجة ثنائية الاختيار ، أى أن التشعب عند التنفيذ يتم باختيار أحد مسارين فقط . و يوضح الشكل التالى هذا الأمر :



و يتكون جزء «الشرط» من :

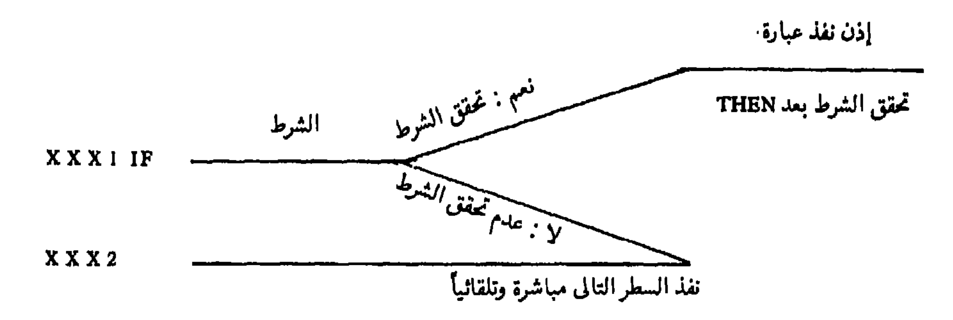
أ ــ المقارن الأول والمقارن الثانى : وهذه تمثل البيانات سواء كان ذلك قيمة أو بما تحتويها أسماء المتغيرات من قيم . أى أن X 1 و X 2 قد تأخذان أرقاماً ثابتة مثل ٦ ، ١٠ ، ٢,٥ .. إلخ ، أو أسماء لمتغيرات (صناديق) مثل M1, K , N ... إلخ . و يشترط أن تخزن قيمة محددة فى المتغير المستخدم فى عبارة IF/THEN قبل تنفيذها .

ب ــ رمز المقارنة : و يستخدم هذا الرمز للمقارنة بين المقارن الأول والثانى ، ويمكن أن يأخذ أحد الرموز التالية :

الرموز المستخدمة فى المقارنة بعبارة IF / THEN	
يساوى	=
أصغر من	<
أكبر من	>
يساوى أو أكبر من	> =
يساوى أو أصغر من	< =
لايساوى	< >

٢ ــ **عبارة تحقق الشرط** : وهى العبارة التى سيتم تنفيذها فى حالة ما إذا كانت نتيجة المقارنة قد تحققت . وإحدى العبارات هذه قد تكون عبارة GOTO X X X أو غيرها . وسنكتفى هنا بالتحدث عن كيفية تنفيذ عبارة IF/THEN إذا أدرجت معها تعليمة GOTO . وفى حالة عدم تحقق الشرط فإن البرنامج سينفذ التعليمة الموجودة فى السطر الذى يلى مباشرة عبارة IF/THEN

و يوضح الشكل التالى هذا التسلسل :



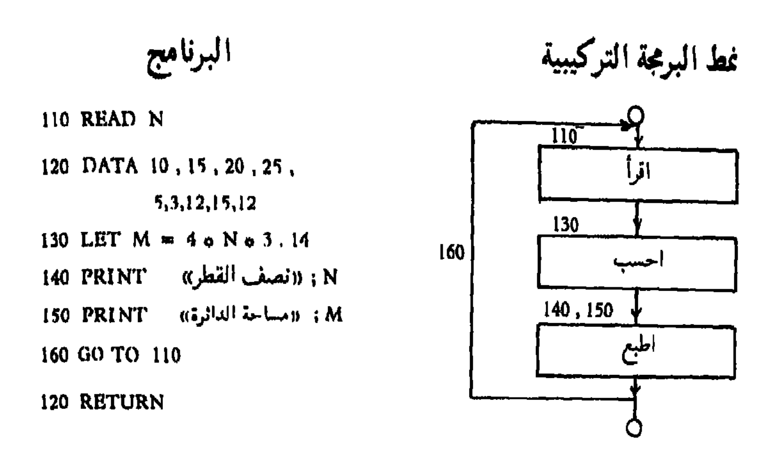
وسنوضح الآن كيفية استخدام عبارة IF / THEN لإيقاف تنفيذ البرنامج بشكل طبيعى ، أى بعد نفاد البيانات .

تطو ير برنامج باستخدام IF / THEN لإيقاف التنفيذ طبيعياً :

إن الوصول إلى معرفة اللحظة التى تتم فيها معالجة آخر قيمة للبيانات ، يمكن تحديده بإحدى وسيلتين : باستخدام قيمة وهمية ، أو بالمعرفة المسبقة لعدد البيانات .

أ ــ وضع قيمة وهمية فى نهاية القيم التى يراد معالجتها، بحيث يوقف تنفيذ البرنامج عند قراءة هذه القيمة الوهمية النهائية ، وقبل تنفيذ الأسطر من البرنامج والتى تؤدى إلى معالجتها . وهذا التوقيف واضح السبب ، إذ أن هذه القيمة الوهمية ، بالتعريف ، هى فقط لتؤدى مهمة معينة وليس لهدف معالجتها .

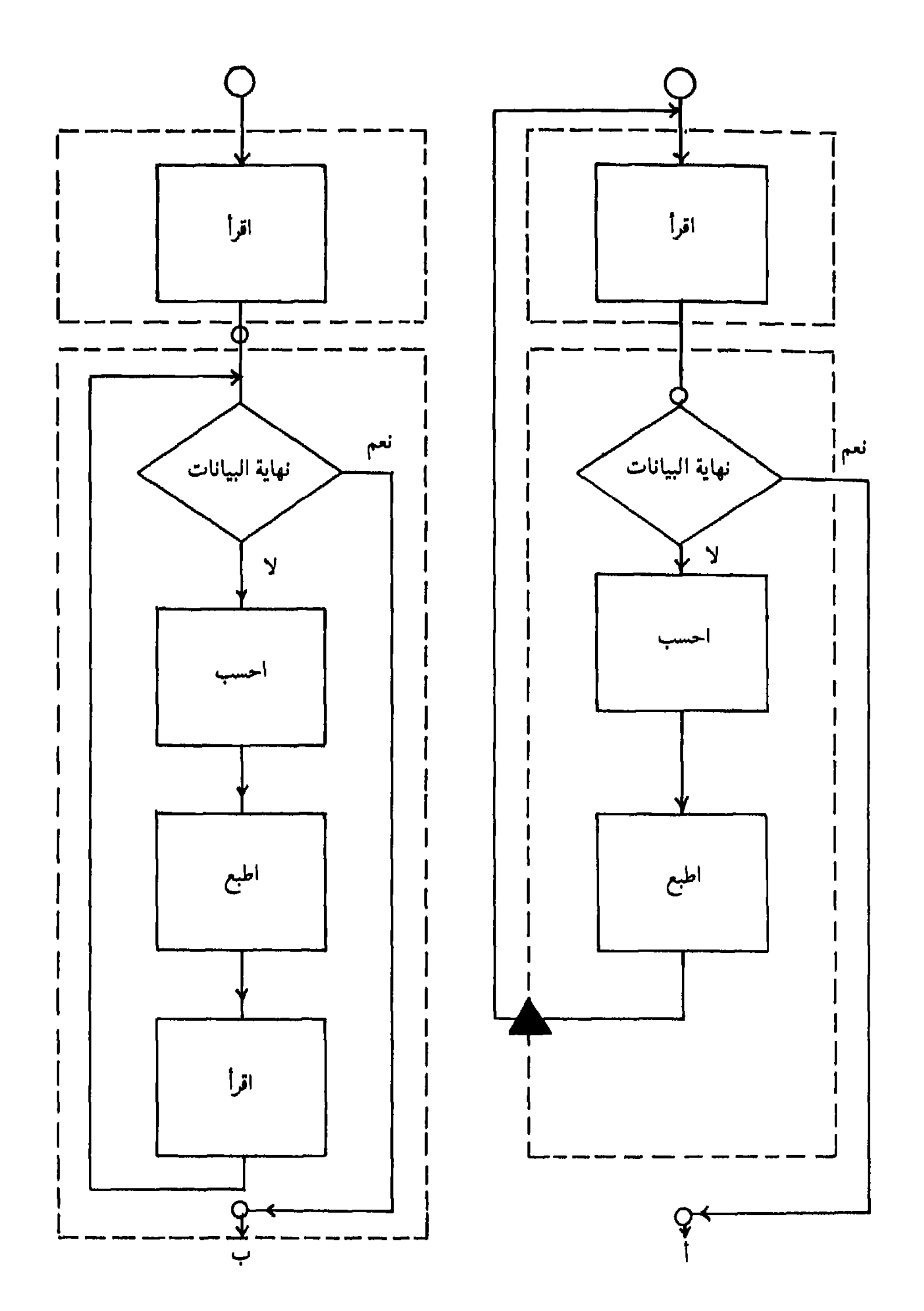
إن الخطوات المعطاة للمثل السابق قد أعيدت هنا لتبيان كيفية تعديلها لتوافق هذا الأسلوب ، وسنرفق مع البرامج نمط البرمجة التركيبية لتتسع التغييرات والاختلافات بين الأساليب المختلفة لإيقاف تنفيذ البرنامج .



فتلاحظ بوضوح كيف أن البرنامج يدخل فى دوارة لانهائية ، وأن الدائرة الصغيرة فى نهاية الشكل لن يحدث تشعب لها . وإذا تمت إضافة قيمة وهمية ، ــ ١ مثلاً ، فى نهاية البيانات فى السطر 20 ، يصبح كالتالى :

120 DATA 10 , 15 , 20 , 25 , 12 , 15 , 12 , = 1

فمن الممكن استخدامها فى عبارة IF/THEN بغرض توقيف البرنامج بعد قراءتها مباشرة . وقبل صياغة عبارة IF/THEN لابد من معرفة موقعها منطقياً فى تسلسل الخطوات . فالقراءة تتم فى الخطوة الأولى والمعالجة فى الثانية ، إذن لابد من ظهور عبارة IF/THEN بينهما . ويصبح الشكل كالتالى ، شكل أ ، وهذا الشكل لايناسب مبادىء البرمجة التركيبية تماماً ؛ لأن تحديد بدايات ونهايات الأجزاء ليس واضحاً ، ومن الممكن تطوير أسلوب أفضل كما فى الشكل ب ، وذلك بزيادة خطوة القراءة مرة أخرى فى النهاية .


اقرأ
نهاية البيانات
نعم
لا
احسب
اطبع
اقرأ
ب
اقرأ
نهاية البيانات
نعم
لا
احسب
اطبع
أ

لاحظ الخطوط المتقطعة التى تحدد الأجزاء ، تجد أنه فى الشكل «أ» يقاطع خط التشعب العائد إلى أعلى عند المثلث المظلل فى حين تم تقسيم الشكل «ب» إلى جزأين متتابعين مستقلين .

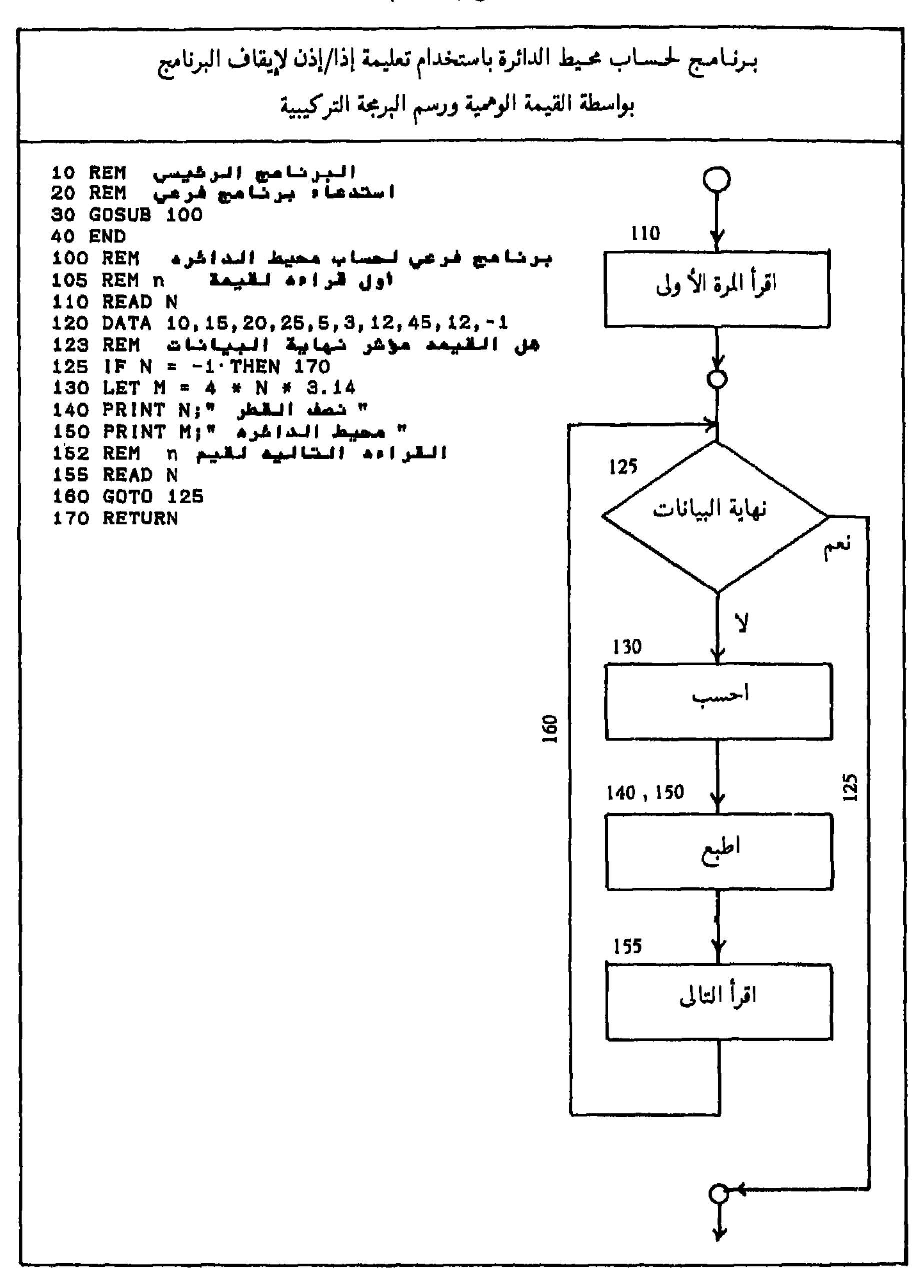
ولتعديل البرنامج السابق المذكور فى شكل (٧ ــ ١) علينا إضافة عبارتى IF/THEN و READ الثانية . ولتحديد رقمى السطرين ، نعود إلى الشكل ب الذى يوضح مسار التشعب حسب أساليب البرمجة الهيكلية . ففى الخطوط الموصلة بين الرموز ، فيما عدا تلك التى توصل بين المربعات المتتالية ، يتم استخدام عبارة GOTO للدلالة على التشعب إلى سطر سابق أو إلى سطر لاحق .

بناء على ذلك فإن الخطوة التى تتضمن عبارة القراءة READ الثانية تتبع خطوة الطبع ، ومن ثم تليها عبارة الذهاب إلى الخطوة التى تقوم بعملية الحساب . أما عبارة IF/THEN فهى تلى عبارة READ الأولى . وعلى وجه التحديد يضاف السطران التاليان للبرنامج :

```
120 IF N = — 1THFN GOTO 170
155 READ N
```

وإليك الآن البرنامج الكامل إلى جانب شكل البرمجة التركيبية للمقارنة بينهما ، بالإضافة إلى تتبع عدد القراءات وما يصاحبها من تغيير فى القيمة المخزنة فى N .

شكل (٧ ــ ٢)



برنامج لحساب محيط الدائرة باستخدام تعليمة إذا/إذن لإيقاف البرنامج بواسطة القيمة الوهمية ورسم البرمجة التركيبية

```
10 REM   البرنامج الرئيسي
20 REM   استدعاء برنامج فرعي
30 GOSUB 100
40 END
100 REM   برنامج فرعي لحساب محيط الدائره
105 REM n   اول قراءه للقيمة
110 READ N
120 DATA 10,15,20,25,5,3,12,45,12,-1
123 REM   هل القيمه مؤشر نهاية البيانات
125 IF N = -1 THEN 170
130 LET M = 4 * N * 3.14
140 PRINT N;" نصف القطر "
150 PRINT M;" محيط الدائره "
152 REM n القراءه التاليه للقيم
155 READ N
160 GOTO 125
170 RETURN
```

مايجرى للقيمة المخزنة داخل الصندوق N	
عداد القراءة	**قيمة N**
الأولى	10
الثانية	15
الثالثة	20
الرابعة	25
⋮	⋮
التاسعة	12
العاشرة (التوقف عن المعالجة)	-1

لاحظ أن قراءة القيمة الجديدة للمتغير N تمحو القيمة السابقة لها ، كذلك فإن آخر قيمة مخزنة فى N هى ١ ـــ ، أى القيمة الوهمية .

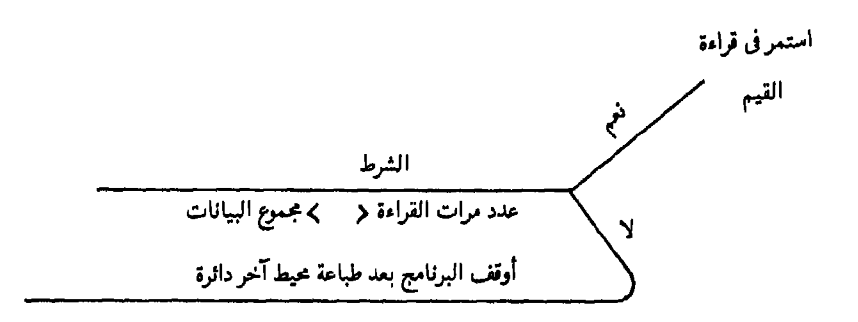
لقد شرحنا فى القسم السابق استخدام القيمة الوهمية فى نهاية قيم البيانات ، وسنتعرض الآن للأسلوب الثانى .

ب ـــ معرفة عدد البيانات المدخلة : لنفترض أننا على علم بأن عدد الدوائر المراد حساب مساحتها هى ٩ ، فبالإمكان التدقيق بعد كل قراءة لمعرفة ما إذا تم قراءة جميع البيانات . وهذا يتطلب استخدام اسم متغير ـــ صندوق ـــ لمتابعة ومراقبة عدد مرات

القراءة . وإليك توضيحاً للخطوات المنطقية المتبعة فى تحديد نقطة نهاية البيانات :

أوقف التنفيذ	استمر فى القراءة	نتيجة الشرط	الشروط		
			عدد القيم المقروءة	رمز المقارنة لا يساوى	عدد مرات القراءة : K
	X	نعم	٩	<>	١
	X	نعم	٩	<>	2
	X	نعم	٩	<>	3
	.	.	.		.
	.	.	.		.
	.	.	.		.
	.	.	.		.
	.	.	.		.
	X	نعم	٩	<>	٧
	X	نعم	٩	<>	٨
X		لا	٩	<>	٩

و يزيد فى التوضيح الشكل التالى :



أما خطوات الحل التفصيلية فتكون كالتالى :

١ ــ اقرأ قيمة نصف القطر.

٢ ــ أضف «١» إلى قيمة العداد السابقة .

٣ ــ احسب المحيط .

٤ ــ اطبع المحيط .

٥ ــ إذا كانت قيمة العداد = ٩ إذن توقف وإلا فاذهب إلى الخطوة الأولى .

وتتطلب الخطوة الثانية استخدام عبارة لم يتم شرحها بعد ، وهى كيفية استخدام تعليمة الإسناد LET مع المتغير العداد COUNTER ، وذلك لمتابعة عدد مرات تنفيذ سطر أو مجموعة أسطر . و يبين الرسم التالى شكل عبارة LET مع المتغير العداد .

استخدام تعليمة LET مع عداد COUNTER				
XX	LET	X1	= X1	+ ١
رقم السطر		القيمة الجديدة	القيمة السابقة	الزيادة فى القيمة
		نفس المتغير العداد COUNTER		

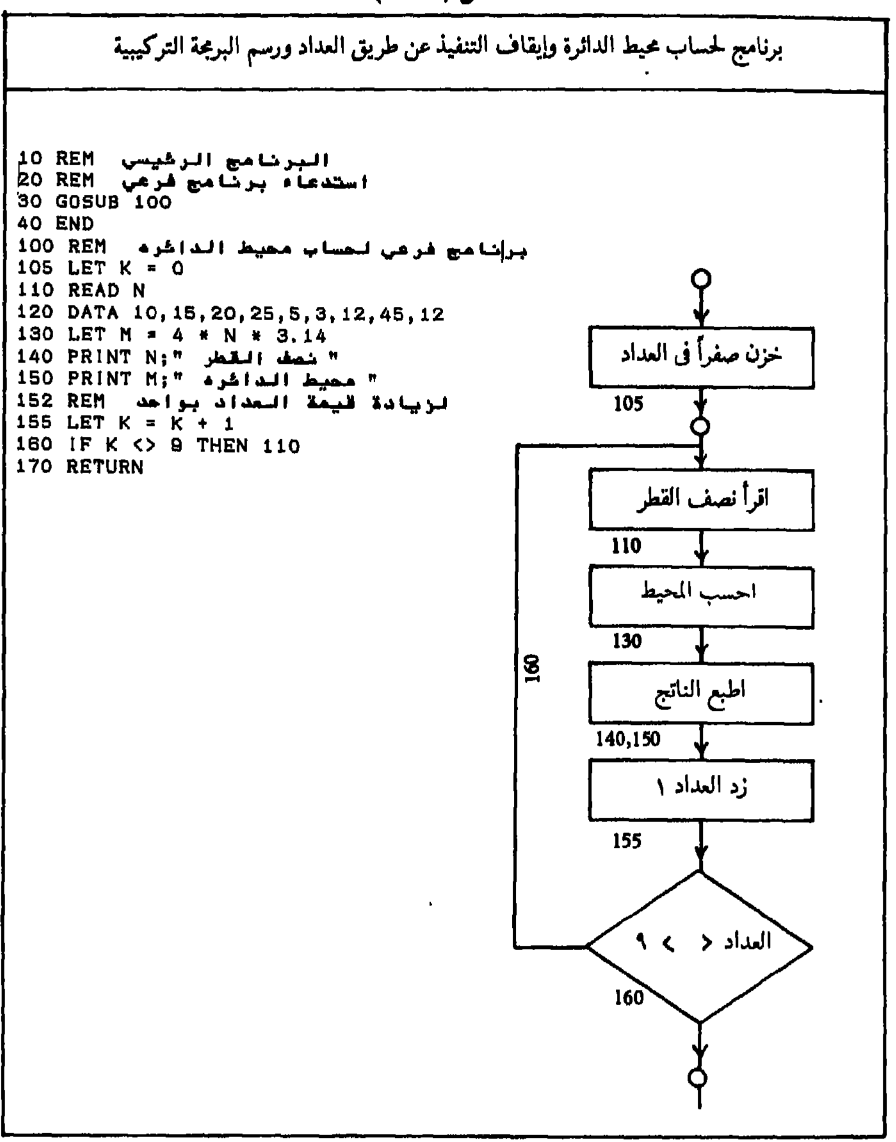
واستخدام إشارة «=» هنا ليس بمضمونها الجبرى ، وإلا كان ١ = صفراً ، بل هو بمعنى البرمجة . أى أن البرنامج يأخذ القيمة السابقة (الحالية) المخزنة فى العداد ــ الصندوق ــ و يضيف واحداً لها ، ومن ثم يعود إلى تخزين القيمة الجديدة فى نفس الصندوق . وإليك توضيحاً لما يجرى داخل الذاكرة ، على فرض أن القيمة السابقة = صفراً .

لحاسب

القيمة الحالية : قبل تنفيذ تعليمة LET
X 1
0

و يوضح الشكل التالى البرنامج المعدل لحساب محيط الدائرة، حسب هذا الأسلوب فى إيقاف تنفيذ البرنامج، مع شكل البرمجة التركيبية المناسب .

شكل (٧ ـ ٣)

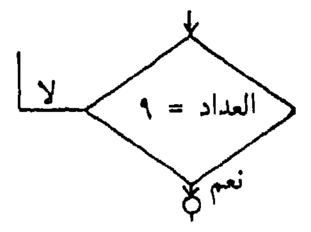


برنامج لحساب محيط الدائرة وإيقاف التنفيذ عن طريق العداد ورسم البرمجة التركيبية

```
10 REM  البرنامج الرئيسي
20 REM  استدعاء برنامج فرعي
30 GOSUB 100
40 END
100 REM  برنامج فرعي لحساب محيط الدائره
105 LET K = 0
110 READ N
120 DATA 10,15,20,25,5,3,12,45,12
130 LET M = 4 * N * 3.14
140 PRINT N;" نصف القطر "
150 PRINT M;" محيط الدائره "
152 REM  لزيادة قيمة العداد بواحد
155 LET K = K + 1
160 IF K <> 9 THEN 110
170 RETURN
```

لاحظ استخدام إشارة< > بدلا من = لتسهيل عملية كتابة البرنامج . ولو أردنا استخدام = لكان شكل البرمجة التركيبية كما فى السابق ، سوى تبديل كلمتى «نعم» و «لا» إحداهما مكان الأخرى . أما البرنامج فسيتغير فيه الأسطر الخاصة بإنهاء تنفيذ البرنامج على الشكل التالى ، و يليه الجزء المعدل من رسم البرمجة التركيبية :

```
160 IF K = 9 GOTO 170
165 GOTO 110
```



لذلك كان الاختلاف فى رسم البرمجة التركيبية لايتغير جوهرياً ، فى حين ساعد استخدام< > بدلاً من = فى السطر 165 فى التخلص من ضرورة استخدام واحدة من عبارات التشعب GOTO .

لاحظ وجود السطر 105 الذى يقوم بوضع قيمة أولية «صفر» فى العداد «K» ، وهذا ينطبق على جميع المتغيرات التى تستعمل كعدادات ، ويجب أن يتم ذلك قبل الوصول إلى مرحلة تنفيذ السطر الذى تظهر فيه عبارة LET مع العداد .

ويختلف شكل ٢ عن شكل ١ بالأسطر التالية فقط :

160 , 155 , 105

ومن الممكن متابعة تسلسل قراءات القيم بطريقة أخرى ، وإن كانت فى جوهرها مشابهة للطريقة السابقة . والاختلاف هو فى توقيت وكيفية تدقيق انتهاء قيم البيانات .

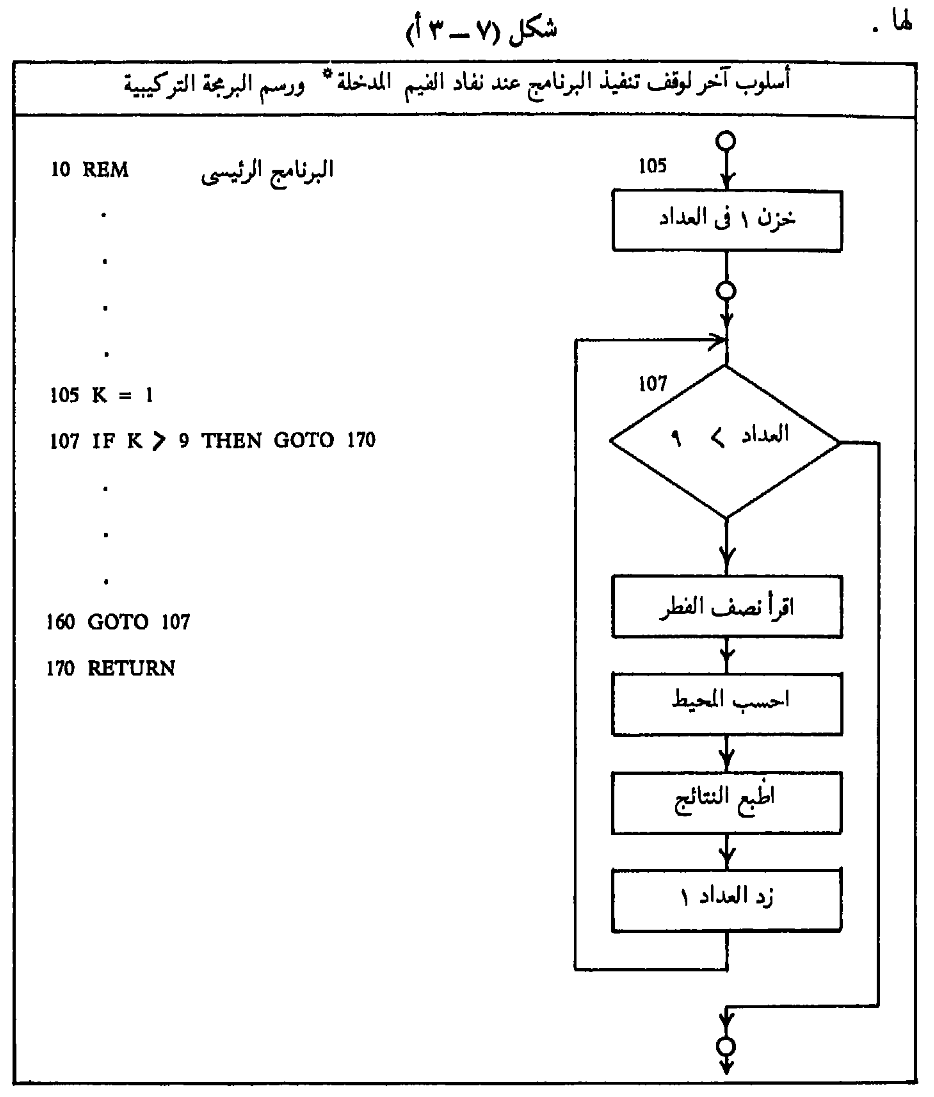
فإذا أردنا تدقيق ما إذا كان العداد يزيد على ٩ ، فإن ذلك يجب أن يتم فى بداية البرنامج وقبل قراءة القيمة التالية لنصف القطر . وذلك لتجنب الخطأ الناتج عن تنفيذ

عبارة القراءة للمرة العاشرة دون وجود بيانات . ولما كان التدقيق يتم قبل القراءة ، فإن القيمة الأولية للعداد يجب أن تكون ١ وذلك ليشير إلى القراءة الأولى .

وإليك البرنامج حسب هذه الطريقة الجديدة مع رسم البرمجة التركيبية المناسب لها .

شكل (٧ ــ ٣ أ)

أسلوب آخر لوقف تنفيذ البرنامج عند نفاد القيم المدخلة* ورسم البرمجة التركيبية



*ذكرت هنا الأسطر المختلفة فقط عن البرنامج السابق

وقد اضطررنا إلى استخدام عبارتين للتشعب غير المشروط GOTO ، وإن كان بالإمكان العودة إلى البرنامج الرئيسى من سطر 107 إذا ظهر على الشكل التالى :

```
107 IF K > 9 THEN RETURN
```

ولكن أساليب البرمجة التركيبية تتطلب التشعب إلى نهاية الجزء . وفى حالة تكرار أماكن العودة إلى البرنامج الفرعى من أسطر مختلفة ، فمن الأفضل الإشارة إلى تعليمة RETURN مرة واحدة ، وذلك منعاً للإرباك ولسهولة تتبع تنفيذ البرنامج . وسيتم الشرح التفصيلى للشكل المذكور سابقاً لتعليمة IF/THEN ، والذى يسمح بإيراد تعليمات أخرى غير الذهاب إلى سطر GOTO بعد تعليمة THEN ، فى الفصل التالى .

تمارين

١ ـ اعمل على طباعة وتنفيذ البرامج الموجودة فى أشكال الفصل ، وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعديلات .

٢ ـ بين الأخطاء إن وجدت فى اختيار أسماء المتغيرات غير العددية التالية :

B%	A5$	$	A
«R 5»	7	P£	B . $

٣ ـ بين الأخطاء إن وجدت فى القيم غير العددية التالية :

«HIS AGE IS 28	«THE NAME IS»
«THE STUDENT'S BOOK»	«P.0 . Box» (404)

٤ ـ اختر أسماء المتغيرات الملائمة للبيانات التالية :

P.0. Box	777 - 7777	75.2
60 METERS	70 KPH	ALI

٥ ـ بين الأخطاء إن وجدت فى كل من العبارات التالية :

أ ـ `10 INPUT 'ENTER NAME , AGE' , N , A`

ب ـ `10 READ N$ , P$ , A , N`

جـ ـ `10 PRINT N, P , R ; S`

د ـ `10 GOTO END`

هـ ـ `10 IF A 5 THEN 40`

٦ ــ بين عدد المرات التى يتم فيها تنفيذ البرامج التالية :

أ ــ

```
10   LET     I = I
20   LET     I = I + I
30   LET     PRINT I
40   IF      I = 5 THEN 20
50   END
```

ب ــ

```
10   LET     K = O
20   READ    A , B
30   PRINT   A , B
40   IF      A = 4 THEN 70
50   DATA    10 , 5 , 20 , 4 , 4 , 8 , 9 , 2
60   END
```

٧ ــ بين نتائج تنفيذ البرامج التالية :

أ ــ

```
10   READ    A , B
20   IF      A C 2 THEN 60
30   PRINT   A + B , A + B
40   DATA    5 , 4 , 2 , 1, 3 1 , 5 , 2 , 1 , 4 , 2 , 6 , 7 , 3
50   COTO    10
60   END
```

ب ــ

```
10   READ    A , B , C
20   IF      A$ = 'AA' THEN 70
```

```
30    PRINT    A$ , B , C
40    DATA     ALT , 70 , 80 , AHMED , 70 , 60 , AAN , 60 , 50
50    DATA     SALIM , 80 , 70 , AA , 60 , 60 , OMAR , 80 70
60    COTO   10
70    END
```

٨ ــ باستخدام تعليمة IF/THEN اكتب برنامجاً يعمل على طباعة الأرقام من عشرة إلى عشرين .

٩ ــ يتم تقويم الدارس فى البرامج الإعدادية فى معهد الإدارة العامة على الأسس التالية :

ــ ألا يقل عدد التقويمات خلال الفصل عن أربعة ، بما فيها التقويم النهائى .

ــ ٣٠٪ للتقويم النهائى .

ــ ألا يزيد وزن التقويم الواحد على ٢٠٪ .

وفى نهاية الفصل التدريبى يتم تحضير جدول تظهر فيه العلامة لكل تقويم من الوزن الخاص بها . والعلامة من ٧٠ لأعمال الفصل ، والعلامة من ٣٠ للتقويم النهائى ، والعلامة النهائية من ١٠٠ لتقويم الفصل ، وأخيراً التقويم للفصل من ٤ (أى يطرح ٥٠ من العلامة النهائية و يقسم الناتج على ١٠ ، فإذا كانت العلامة ٦٠ يكون التقويم ١ و ل ٧٠ ، ٢ و ل ٨٠ ، ٣ و ل ٩٠ ، ٤ وهكذا أيضا للعلامات الواقعة فيما بينها) .

طور برنامجاً لطباعة الجدول ، إذا أراد المدرب إدخال العلامات من ١٠٠ وأوزانها المرادفة للتقاويم الفصلية وعلامة التقويم النهائى . مع العلم بأن القوانين لا تسمح بالكسور العشرية فى العلامات . ماعدا التقويم النهائى من ٤ .

الفصل الثامن

تعليمات التشعب

مقدمة عن ضرورة التشعب لمعالجة البيانات :

تحدثنا فى الفصل السابق عن التشعب الناتج عن انتهاء البيانات المدخلة ، والذى يستخدم تعليمة التشعب المشروط بشكلها المبسط IF/THEN مقرونة بتعليمة التشعب غير المشروط GOTO لتجاهل (القفز عن) تنفيذ مجموعة من التعليمات ، وكان التدقيق يتم إما عن طريق قيمة وهمية تظهر فى آخر البيانات أو عن طريق المعرفة المسبقة بعدد البيانات المدخلة .

ولكن كثيراً من التطبيقات العملية فى المجالين التجارى والحكومى ، تتطلب تفرعات مشروطة وغير مشروطة متعددة بخلاف ما تقدم . ومن أنواع هذه التفرعات : التفرع بين وظائف الأجزاء المتعددة ، والتفرع داخل الجزء الواحد .

١ ـ التفرع من الأجزاء : أوضحنا فى الفصل الثالث أمثلة من هياكل هرمية واقعية ، و يتطلب تنفيذ البرنامج المطور لأى تركيب هرمى الانتقال من وظيفة إلى أخرى ، أى من جزء إلى آخر . و بشكل رئيسى التفرع العمودى من الجزء المتحكم إلى الأجزاء المنفذة التابعة له ، وخاصة إذا كان التشعب يستدعى اختياراً واحداً أو أكثر من الأجزاء بناء على المقارنة بين قيم أحد المتغيرات .

لنفرض أن المتغير TYP يحدد أياً من الأجزاء التنفيذية الثلاثة التالية ، إذن يجب تنفيذ X 1 , X 2 , X 3 .فإذا كانت قيمة TYP = 1 يتم تنفيذ الجزء X 1 ، وإذا كانت قيمة TYP = 2 يتم تنفيذ الجزء X 2 ، وإذا كانت قيمة TYP = 3 يتم تنفيذ الجزء X 3 .

و يقوم البرنامج الرئيسى بتوجيه التنفيذ حسب القيمة المخزنة فى TYP فى أى مرحلة من مراحل تنفيذ البرنامج .

٢ ـ التفرع داخل الأجزاء التنفيذية : قد يتطلب الأمر التفرع أثناء تنفيذ الجزء لوظيفته المحددة . فالجزء الذى يحسب الراتب الإجمالى يكون التفرع بناء على عدد ساعات العمل ، وذلك إذا اقتضى الأمر حساب خارج الدوام . والجزء الخاص بحساب بدلات النقل يكون التفرع فيه حسب مرتبة الموظف ... وهكذا . ففى هذه الحالات يتم التفرع بناء على تدقيق قيم متغيرات مدخلة ، وغالباً ما تتم هناك معالجات حسابية ، وطباعة مخرجات ، وقراءة مدخلات مختلفة باختلاف حالة التفرع .

وهناك حالات أخرى تتطلب التفرع ، وهى حالات التصنيف والفرز والتبو يب والتلخيص للبيانات المدخلة . ومن أمثلة هذه الحالات : حساب إجمالى مبيعات المناطق الثلاث الشرقية والوسطى والغربية لشركة سيارات ، تصنيف الدرجات إلى A , B , C , D , F أى إلى امتياز ، جيد جداً ، جيد ، مقبول ، وراسب على التوالى ، تبو يب الأعمار إلى فئات متتالية أقل من ١٦ ، ١٦ ــ ٢٠ ، ٢١ ــ ٢٥ ، ٢٦ ــ ٣٠ ، ٣١ ــ ٣٥ ، ٣٦ ــ ٤٠ ، ٤١ ــ ٤٥ ، وأكبر من ٤٥ .

وكذلك فإن استخلاص بعض المعلومات من ملف البيانات المدخلة وفق شروط محددة ، يتطلب تشعباً متعدداً . ومن الأمثلة على ذلك تحضير قائمة بأسماء الموظفين التابعين للدائرة المالية أو الدائرة الإدارية ، ورواتبهم تزيد على ٨٠٠٠ ريال شهرياً ، أو معرفة أسماء العمال الذين يعملون فى إنتاج القطعة الأ ولى أو الثانية ، ولهم سنوات خبرة تزيد على خمس سنوات ، أو توزيع المهام حسب المستوى الأكاديمى أو عدد سنوات الخبرة والعمر... وهكذا .

وكلا النوعين المذكورين سابقاً تتطلب معالجتهما استخدام تعليمة التشعب المشروط IF/THEN المتعددة ، وتعليمات أخرى سيتم تقديمها فى هذا الفصل .

وسنبحث استخدام هذه التعليمات فى حالات التشعب داخل الجزء المنفذ أولاً ، ومن ثم التشعب من الجزء المتحكم إلى الأجزاء المنفذة . والنوع الأ ول يمثله التشعب الثنائى ، ويمثل النوع الثانى التشعب المتعدد .

التشعب الثنائى

تطو ير برنامج باستخدام تعليمة IF / THEN :

مثال (٨ ــ ١) :

الهدف : إيجاد الراتب الإجمالى لكل موظف فى مؤسسة معينة .

أولاً ــ خطوات الحل :

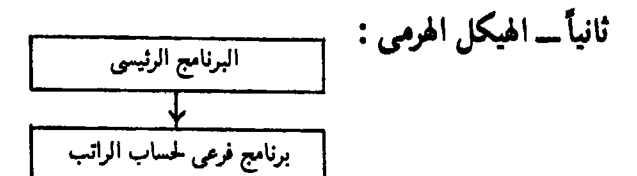
١ ــ الحصول على اسم الموظف ، عدد ساعات العمل الأسبوعية وأجرة الساعة .

٢ ــ إيجاد المبلغ المستحق لخارج الدوام إن وجد ، والمبلغ الإجمالى .

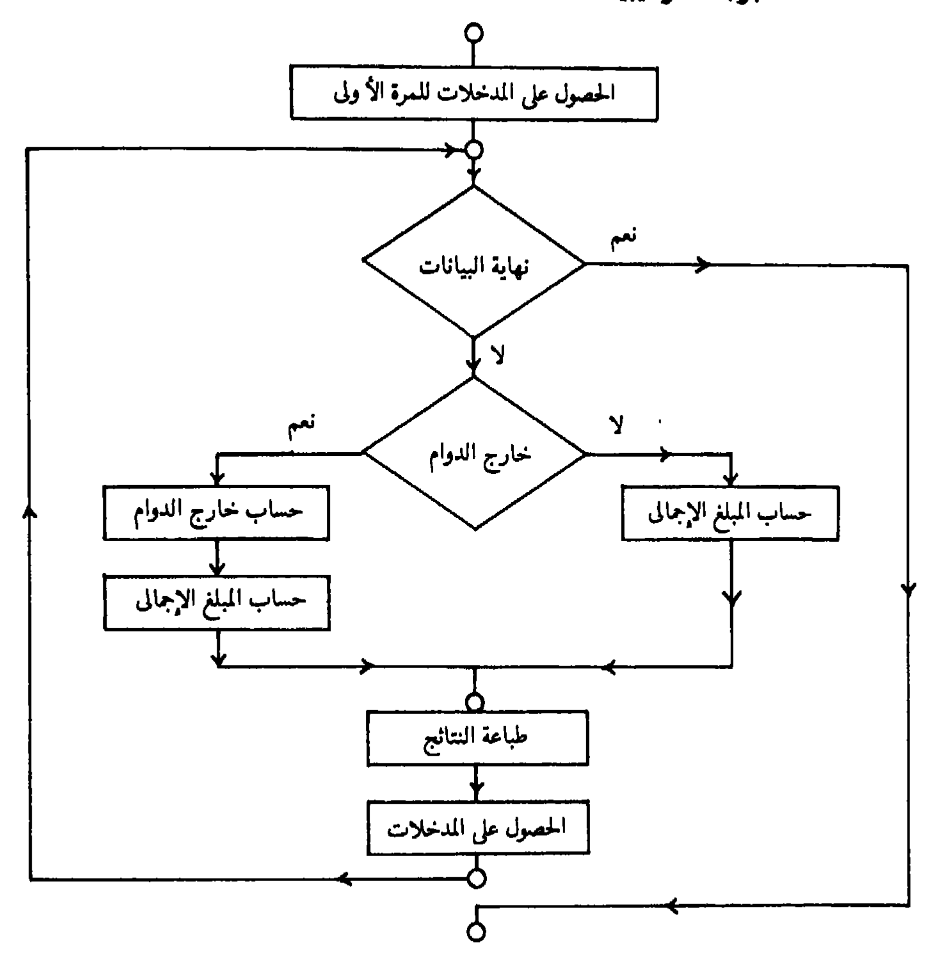
٣ ــ طباعة اسم الموظف ، عدد ساعات العمل ، الراتب الأساسى ، المبلغ المستحق من خارج الدوام والمبلغ الإجمالى .

وسيتم تطو ير برنامج لتحقيق هذا الهدف يحتوى على برنامج فرعى واحد يستخدم فيه تعليمة IF/THEN مقرونة بتعليمة GOTO كما تم شرحه فى الفصل السابق .

ثانياً ــ الهيكل الهرمى :



ثالثاً ــ نمط البرمجة التركيبية :



لاحظ التشعب عند تدقيق خارج الدوام ، ومن ثم الاكتفاء بطباعة النتائج والحصول على المدخلات للمرات المتتالية .

رابعاً ــ ويحوى الشكل التالى البرنامج كاملاً حسب الخطوات المبينة سابقاً .

شكل (٨ ــ ١)

برنامج لحساب الراتب الإجمالى لموظفى مؤسسة معينة باستخدام عبارة IF/THEN واحدة

```
10 REM لا ستدعاء البرنامج الفرعي
20 GOSUB 50
30 END
40 REM  البيانات
50 DATA "ALI AHMED",35,20
60 DATA "SALIM ALI",42,25
70 DATA "MOHAMED OMER",60,19
80 DATA "AHMED HASAN",52,26
90 DATA "dummy",0,0
100 REM  لطباعة العناوين
110  PRINT  "    NAME           HOURS WORKED      RATE    BASE SALARY  OVERTIME  N
ET PAY"
120  PRINT  "===============  ============      ====    ===========  ========  =
======"
130 REM لقراءة اسم الموظف ،عددساعات العمل ،اجرة الساعه
140 READ  N$,H,P
150 REM   للا ستفسار عن نهاية البيانات
160 IF  N$ ="dummy"  THEN  350
170 REM  للا ستفسار عن وجود خارج دوام
180 IF  H <= 40 THEN 250
190 REM   ايجاد عدد ساعات الدوام والمبلغ المستحق لها
200 LET  O = ((H - 40) * 1.5) * P
210 REM   لا يجاد المبلغ الا جمالي
220 LET  N = (40 * P) + O
230 GOTO 280
240 REM   لا يجاد راتب الشخص الذى لم يعمل خارج دوام
250 LET  N = H * P
260 LET  O = 0
270 REM لطباعة اسم الموظف ،عددساعات العمل ،اجرة الساعه ،خارج دوام ،راتب اجمالي
280 REM  للا يجاد الراتب الا ساسي
290 LET B = N - O
300 PRINT TAB(2);N$; TAB(20);H; TAB(37);P; TAB(48);B; TAB(58);O;TAB(67);N
310 PRINT  "------------------------------------------------------------------
-----"
320 REM لقراءة اسم الموظف ،عددساعات العمل ،اجرة الساعه
330 READ  N$,H,P
340 GOTO 160
350 RETURN
```

وعند تنفيذ البرنامج ستظهر النتائج كالتالى :

NAME	HOURS WORKED	RATE	BASE SALARY	OVERTIME	NET PAY
ALI AHMED	35	20	700	0	700
SALIM ALI	42	25	1000	75	1075
MOHAMED OMER	60	19	760	570	1330
AHMED HASAN	52		1040	468	1508

فى البرنامج السابق نلاحظ وجود أربع عبارات تستخدم تعليمة اذهب إلى GOTO ، وهذا مخالف إلى حد ما لقواعد البرمجة التركيبية .

لذا يمكن تعديله وكتابته بطريقة أفضل باستخدام شكل آخر لعبارة :

IF / THEN

فلإيجاد :

١ ــ راتب الموظف الذى لم يشتغل خارج دوام ، يمكننا عمل ذلك باستخدام العبارة التالية :

```
150 IF H <= 40 THEN N = H * P
```

لتحل مكان العبارات السابقة (150 , 220 , 230) .

٢ ــ راتب الموظف الذى يوجد له خارج دوام ، تستخدم العبارة التالية :

```
170 IF H > 40 THEN O = (H - 40) * (P * 1 . 5)
```

ومن ثم نجد الراتب الإجمالى للموظف الذى يوجد له خارج دوام بإضافة خارج الدوام إلى الراتب الأساسى :

```
190 IF H > 40 THEN N = O + (P * 40)
```

وبإجراء هذه التعديلات تصبح الصيغة النهائية للبرنامج كالتالى :

شكل (٨ ـ ١ أ)

برنامج لحساب الراتب الإجمالى لموظفى مؤسسة معينة باستخدام عبارات IF / THEN متعددة

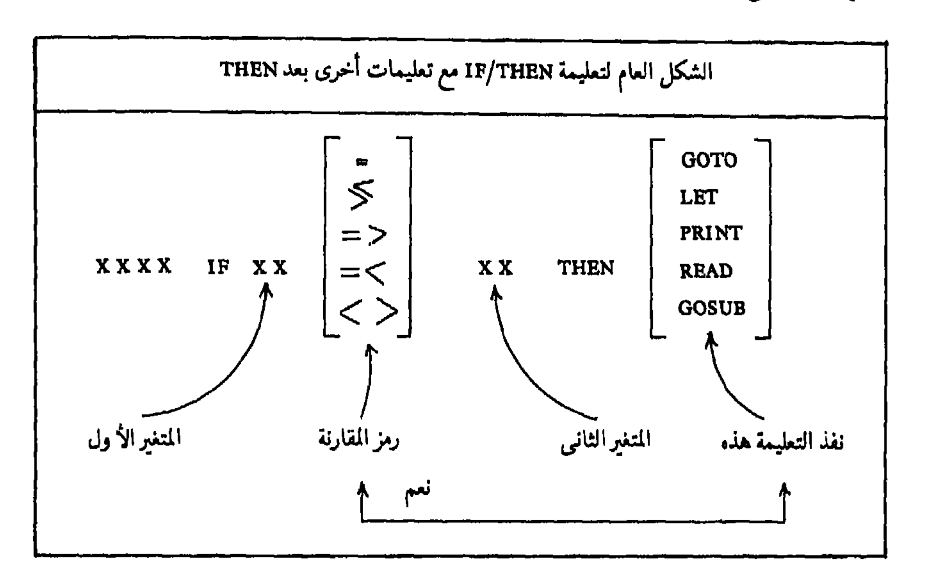
```
10 REM لا ستدعاء البرنامج الفرعي
20 GOSUB 50
30 END
50 REM لطباعة العناوين
60  PRINT "    NAME           HOURS WORKED      RATE    BASE SALARY  OVERTIME  NE
T PAY"
70  PRINT "===============  ============      ====    ===========  ========  ==
====="
80 REM لقراءة اسم الموظف ،عددساعات العمل ،اجرة الساعه
90 READ  N$,H,P
100 REM  للاستفسار عن نهاية البيانات
110 IF  N$ ="dummy"  THEN  350
115 LET O = 0
120 REM  للاستفسار عن وجود خارج دوام
130 IF  H <= 40 THEN N = H * P
140 REM  ايجاد عدد ساعات الدوام والمبلغ المستحق لها
150 IF H > 40 THEN  O = (H - 40) * ( 1.5 * P)
160 REM  لايجاد المبلغ الاجمالي
170 IF H > 40 THEN  N = (40 * P) + O
220 REM لطباعةاسم الموظف ،عددساعات العمل ،اجرةالساعه ،خارج دوام ،راتب اجمالي
230 REM  للايجاد الراتب الاساسي
240 LET B = N - O
250 PRINT TAB(2);N$; TAB(20);H; TAB(37);P; TAB(48);B; TAB(58);O;TAB(67);N
260 PRINT  "--------------------------------------------------------------------
-----"
270 REM لقراءة اسم الموظف ،عددساعات العمل ،اجرة الساعه
280 READ  N$,H,P
290 GOTO 110
300 DATA "ALI AHMED",60,23
310 DATA "SALIM ALI",42,25
320 DATA "MOHAMED OMER",36,19
330 DATA "AHMED HASAN",52,18
340 DATA "dummy",0,0
350 RETURN
```

وفى حال تنفيذ البرنامج ، تظهر النتائج كالتالى :

NAME	HOURS WORKED	RATE	BASE SALARY	OVERTIME	NET PAY
ALI AHMED	60	23	920	690	1610
SALIM ALI	42	25	1000	75	1075
MOHAMED OMER	36	19	684	0	684
AHMED HASAN	52	18	720	324	1044

لاحظ فى كل من العبارتين (150 , 170) أنه قد تم إتباع تعليمة IF/THEN بعملية حسابية ، بدلاً من إتباعها برقم العبارة التى تجرى تلك العملية الحسابية ، و بذلك تم الاستغناء عن العديد من تعليمات GOTO ، وهذا مايتوافق مع أساليب البرمجة التركيبية .

و يعكس الرسم التالى الشكل العام لتعليمة IF/THEN عند استخدام تعليمات أخرى فى نفس العبارة .



الشكل العام لتعليمة IF/THEN مع تعليمات أخرى بعد THEN

أى أن البرنامج ينفذ التعليمة الظاهرة عقب كلمة THEN فى حالة كون المقارنة صحيحة. أما فى حالة كون المقارنة غير صحيحة ، فيتم تنفيذ السطر التالى تلقائياً .

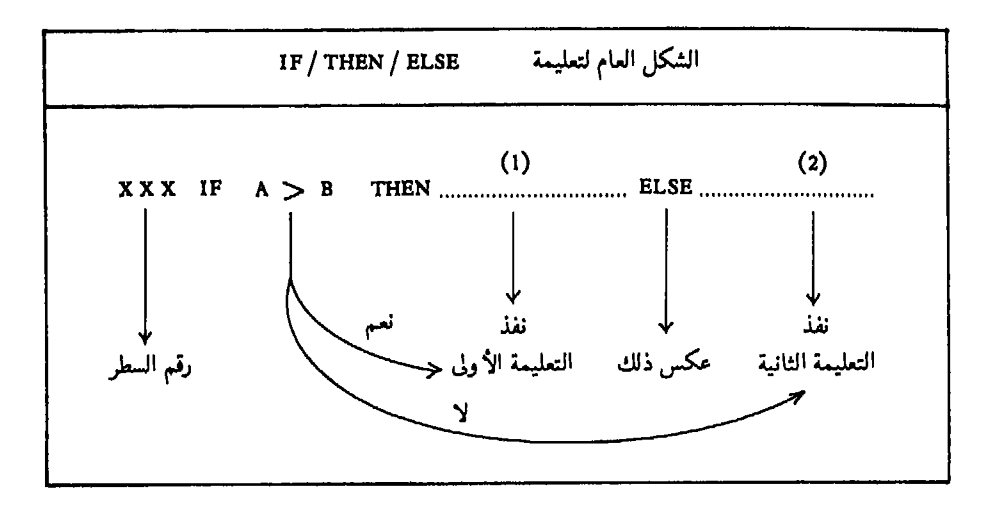
تم تعديل البرنامج السابق ، شكل (٨ ــ ١ أ) بتقليل عدد تعليمات اذهب إلى GOTO ليتناسب مع قواعد البرمجة التركيبية وذلك بإجراء العملية اللازم القيام بها عند تحقيق الشرط ، ولذا زاد عدد المرات التى استخدمت بها تعليمة إذا / إذن IF/THEN ، حيث قمنا بالاستفسار عن كل حالة منفردة دون تغيير منطق البرنامج .

تطوير برنامج باستخدام تعليمة IF/THEN/ELSE :

وسنتعرف الآن على كيفية التقليل من عدد تعليمات إذا/إذن وذلك باستخدام تعليمة إذا / إذن / عكس ذلك

IF / THEN / ELSE

كما هو موضح فى الشكل التالى :



هنا يتم تنفيذ التعليمة الأولى (١) فى حالة تحقيق الشرط ، وهو أن تكون قيمة المتغير A أكبر من قيمة المتغير B ، ولكن إذا لم تكن قيمة المتغير A أكبر من قيمة المتغير B يتم تنفيذ التعليمة الثانية (٢) . وفى كلتا الحالتين يتم تنفيذ تعليمة السطر الذى يليه تعليمة إذا .

ونشير هنا إلى إمكانية وجود أكثر من تعليمة بعد كلمة THEN وكلمة ELSE بشرط أن يفصل بينهما «:» وذلك كما هو موضح أدناه :

10 IF A > B THEN التعليمة الأولى : التعليمة الثانية ELSE

لاحظ أن عـدد الحـروف الـتـى يمـكن أن تظهر فى رقم سطر واحد هو ٢٥٦ حرفاً ، وهـذا يحـد مـن عـدد التعليمات التى يمكن أن تظهر فى رقم سطر واحد . لذلك فى حالة مثل هذه يفضل تقليل الفراغ بين التعليمات قدر الإمكان .

ويمكن تعديل البرنامج السابق وفقاً لهذا الأسلوب كالتالى :

أ) إذا كـان عـدد سـاعـات العمل أقل أو يساوى ٤٠ ساعة فذلك يعنى أن الموظف لم يعمل خارج دوام، وعليه يكون :

خارج الدوام = صفراً

```
LET    O = ϕ
```

والراتب الإجمالى = عدد ساعات العمل × أجرة الساعة

```
LET    N = H * P
```

ب) أما إذا كان عكس ذلك (ELSE) أى لم يتحقق الشرط ، فيعنى أن الموظف قد عمل خارج دوام ، وعليه يكون :

خارج الدوام = (عدد ساعات العمل ــ ٤٠) × (أجرة الساعة × ١,٥)

```
LET    O = (H - 40) * (P * 1.5)
```

والراتب الإجمالى = خارج الدوام + (٤٠ × أجرة الساعة)

```
LET    N = O + (40 * P)
```

يمكننا تنفيذ الخطوتين أ ، ب باستخدام العبارة التالية :

أ)
```
IF    H <= 40 THEN O = O : N = H * P
ELSE
```

ب)
```
O = (H - 40) * (P * 1.5) : N = O + (40 * P)
```

وبإجراء هذه التعديلات ، تصبح صيغة البرنامج كالتالى :

شكل (٨ ــ ١ ب)

برنامج لحساب الرواتب الإجمالية لموظفى مؤسسة معينة باستخدام تعليمة IF / THEN / ELSE

```
10 REM لا ستدعاء البرنامج الفرعي
20 GOSUB 40
30 END
40 REM لطباعة العناوين
50 PRINT "    NAME          HOURS WORKED       RATE     BASE SALARY   OVERTIME   NET
 PAY"
60 PRINT "===============  ============       ====     ===========   ========   ===
===="
70 REM  اسناد قيم اوليه للمتغيرات
80 LET TY = 0: LET T1 = 0
90 REM لقراءة اسم الموظف ،عددساعات العمل ،اجرة الساعه
100 READ  N$,H,P
110 REM  للاستفسار عن نهاية البيانات
120 IF  N$ ="dummy"  THEN  300
130 REM  للاستفسار عن وجود خارج دوام
140 IF  H <= 40 THEN N = H * P : O = O
                ELSE O = (H - 40 ) * (P * 1,5) : N = O + ( P * 40)
150 REM لطباعة اسم الموظف ،عددساعات العمل ،اجرة الساعه ،خارج دوام ،راتب اجمالى
160 REM  للايجاد الراتب الاساسي
170 LET B = N - O
180 LET T = T + N
190 LET.T1 = T1 + O
200 PRINT TAB(2);N$; TAB(20);H; TAB(37);P; TAB(48);B; TAB(58);O;TAB(67);N
210 PRINT  "-------------------------------------------------------------------
-----"
220 REM لقراءة اسم الموظف ،عددساعات العمل ،اجرة الساعه
230 READ  N$,H,P
240 GOTO 120
250 DATA "ALI AHMED",60,23
260 DATA "SALIM ALI",42,25
270 DATA "MOHAMED OMER",36,19
280 DATA "AHMED HASAN",52,18
290 DATA "dummy",0,0
300 PRINT TAB(2);"================================================================
===============" 
310 REM  لطباعة اجمالي خارج الدوام و اجمالي الراتب للموظفين
320 PRINT TAB(40);"t o t a l s";TAB(57);T1;TAB(66);T
330 PRINT TAB(2);"=================================================================
==============="
340 RLTURN
```

وفى حالة تنفيذ البرنامج فى الشكل السابق ستكون المخرجات على الشكل التالى :

NAME	HOURS WORKED	RATE	BASE SALARY	OVERTIME	NET PAY
ALI AHMED	60	23	920	690	1610
SALIM ALI	42	25	1000	75	1075
MOHAMED OMER	36	19	684	0	684
AHMED HASAN	52	18	720	324	1044
		t o t a l s		1089	4413

نلاحظ فى هذا البرنامج كيفية استخدام أكثر من عبارة فى نفس السطر باستخدام الرمز «:» ، كما هو مبين فى العبارة 110 بالإضافة إلى استخدام تعليمة (عكس ذلك ELSE) للتخلص من الاستخدام المتعدد لتعليمة إذا/إذن IF/THEN .

ملخص ماتقدم عن استخدامات تعليمة IF/THEN :

لقد أمكن التحكم فى سير تنفيذ البرنامج باستخدام تعليمة إذا/إذن IF/THEN فى عدة طرق وهى :

١ ـ تحويل سير التنفيذ فى حالة تحقيق الشرط إلى رقم عبارة فى البرنامج

XXX	IF	....	THEN	X X X
↓		↓		↓
رقم السطر		condition / شرط		رقم عبارة

٢ ـ تنفيذ تعليمة (إدخال / إخراج / معالجة) فى حالة تحقيق الشرط .

X X X	IF	....	THEN	....
↓		↓		↓
رقم السطر		condition / شرط		إجراء عملية

٣ ـ تنفيذ عملية (أ) فى حالة تحقيق الشرط وإلا (عكس ذلك) تنفيذ عملية (ب) .

X X X	IF	THEN	ELSE	....
↓	↓	↓		↓
رقم السطر	condition / شرط	نفذ تعليمة (١)		نفذ تعليمة (ب) يمكن ذلك

٤ ـ تنفيذ أكثر من عملية فى عبارة واحدة فى حالة تحقيق الشرط وإلا (عكس ذلك) نفذ العمليات التالية فى العبارة التالية لتعليمة ELSE

XXX	IF	THEN........................	: ....
↓	↓	↓	↓
رقم السطر	شرط	نفذ التعليمة ١	نفذ التعليمة ٢
ELSE	....	:	: ....
↓	↓	↓	↓
وإلا	نفذ تعليمة ١	نفذ تعليمة ٢	نفذ تعليمة ٣

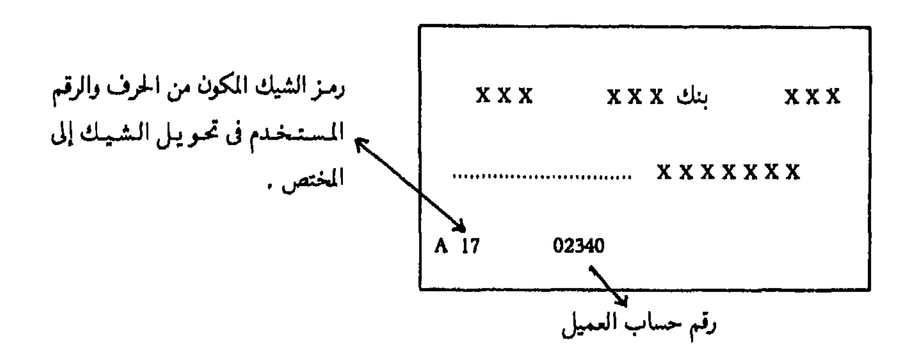
تطوير برنامج باستخدام تعليمة IF/THEN والمقارنات OR , AND :

تم استخدام تعليمة IF/THEN فيما سبق فى وضع يتم فيه التحكم بمسار تنفيذ خطوات البرنامج ، بناء على نتيجة مقارنة أحادية بين قيمتى متغيرين ، ولكن فى بعض الحالات قد تعتمد نتيجة التشعب على تحقيق أكثر من شرط (مقارنة) بين أكثر من زوج من المتغيرات .

ففى هذه الحالة تظهر هذه الشروط (المقارنات) بين كلمتى IF وTHEN وسنعطى هنا مثالاً على كيفية استخدام عبارة IF/THEN مع المقارنات المتعددة لتحقيق التشعب إلى التعليمة الظاهرة بعد THEN

مثال (٨ ــ ٢) :

يعمل بنك معين على تسلم الشيكات ، ومن ثم توزيعها على ثلاثة موظفين بناء على رمز مكتوب على هذه الشيكات ، و يظهر هذا الرمز قبل رقم حساب الزبون ، و يتكون هذا الرمز من جزأين ، الأول : حرف يتكون من أحد الأحرف التالية : B , C , D , A ، والثانى : رقمى مكون من خانتين وذلك حسب الشكل التالى :



(أ) فإذا كان الحرف الأول يساوى أياً من الرموز التالية B , A أو C يتم تحويله للموظف x وذلك بغض النظر عن الجزء الرقمى .

(ب) إذا كان الحرف الأول يساوى D والرقم الذى يليه يساوى 14 يتم تحويله للموظف Y .

(جـ) إذا كان الحرف الأول يساوى D والرقم الذى يليه لايساوى 14 يحول للموظف x .

(د) إذا كان الحرف الأول لايساوى C , B , A , أو D يعتبر هذا الشيك غير سليم ، وتتم إعادته إلى المصدر .

الهدف : الحصول على تقرير فى نهاية الدوام يحتوى على التالى :

١ ــ عدد الشيكات المحولة للموظف X

٢ ــ عدد الشيكات المحولة للموظف y

٣ ــ عدد الشيكات غير السليمة (المعادة) .

أولاً ــ خطوات الحل :

١ ــ الحصول على رمز الشيك والرمز الذى يليه .

٢ ــ الاستفسار عن رمز الشيك إذا كان يساوى A أو B أو C أضف واحداً للعداد الخاص بالموظف X .

٣ ــ الاستفسار عن رمز الشيك إذا كان يساوى D والرمز الذى يليه يساوى 14 أضف واحداً للعداد الخاص بالموظف Y .

٤ ــ الاستفسار عن رمز الشيك إذا كان يساوى D والرمز الذى يليه لايساوى 14 أضف واحداً للعداد الخاص بالموظف X .

٥ ــ الاستفسار عن رمز الشيك إذا كان لايساوى A أو B أو C أو D أضف واحداً للعداد الخاص بالشيكات غير السليمة (المعادة) .

نلاحظ أن كلاً من الخطوات ٢ ، ٣ ، ٤ ، ٥ ، يتطلب منا القيام بأكثر من عملية مقارنة . وسنتعرض الآن للتعليمات الممكن استخدامها فى لغة بيسك لمثل هذا النوع من عمليات المقارنة والمستخدمة فى تعليمة IF ... THEN

١ ـ تعليمة «أو» (OR) وتستخدم للاستفسار عن تحقيق أى من المقارنات ، كما هو مطلوب فى الخطوة ٢ ، حتى تتم زيادة عداد الموظف × بواحد ، وتظهر هذه التعليمة كالتالى :

الشكل العام لتعليمة «أو» (OR)										
XXX	IF	X	=	XX	OR	X	=	XX	THEN	XXXX
↓		↓	↓	↓	↓	↓	↓	↓	↓	↓
رقم السطر		اسم المتغير	رمز المقارنة	قيمة	تعليمة أو	اسم المتغير	رمز المقارنة	قيمة	إذن	تعليمة اقرأ أو اطبع أو أسند أو الذهاب إلى رقم عبارة أو استدعاء برنامج فرعى .

وعليه فيمكننا تمثيل الخطوة (٢) بالعبارة التالية :

XXX IF A$ = «A» OR A$ = «B» OR A$ = «C» THEN N1 = N1 + 1

حيث نضيف واحداً للعداد N1 فى حالة أن يكون المتغير A$ يساوى أياً من القيم «A» أو «B» أو «C»

إذن فى حالة استخدام تعليمة OR يجب تحقيق واحد فقط من الشروط (عمليات المقارنة) حتى يتم تنفيذ تعليمة مابعد THEN .

٢ ـ تعليمة «و» (AND) وتستخدم للتأكد من تحقيق جميع عمليات المقارنة ، كما هو مطلوب فى الخطوة ٣ حتى تتم زيادة عداد الموظف Y بواحد ، وشكل هذه التعليمة كالتالى :

الشكل العام لتعليمة «و» AND											
XXX	IF	X	=	XX	AND	X	=	XX	THEN	XXXXX
↓		↓	↓	↓	↓	↓	↓	↓		↓
رقم السطر		اسم المتغير	رمز المقارنة	قيمة المتغير	تعليمة و	اسم المتغير	رمز المقارنة	قيمة المتغير		عملية أو أمر اذهب إلى رقم عبارة أو أمر استدعاء برنامج فرعى

* يمكن أن يكون رمز المقارنة أياً من الرموز التى تم شرحها

وعليه يمكننا تمثيل الخطوة (٣) بالعبارة التالية :

```
IF   A$ = «D» AND = 14 THEN N2 = N2 + 1
```

أى نضيف واحداً للعداد N2 فى حالة أن يكون المتغير A$ مساو ياً N للقيمة D وقيمة المتغير B تساوى 14

إذن فى حالة استخدام تعليمة «و» (AND) يجب تحقيق جميع الشروط ، (عمليات المقارنة) حتى يتم تنفيذ تعليمة ما بعد كلمة THEN

٣ ـ ويمكننا تمثيل كل من :

أ) الخطوة (٤) بالعبارة التالية :

```
IF   A$ = «D» AND B < > 14 THEN N1 = N1 + 1
```

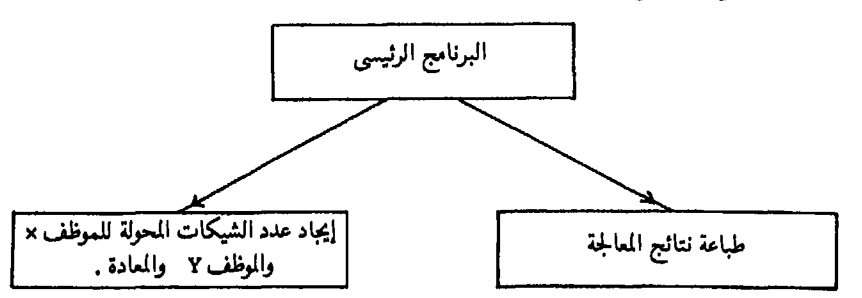
أى إضافة واحد للعداد N1 فى حالة أن يكون المتغير A$ يساوى D والمتغير B لايساوى 14

ب) الخطوة (٥) بالعبارة التالية :

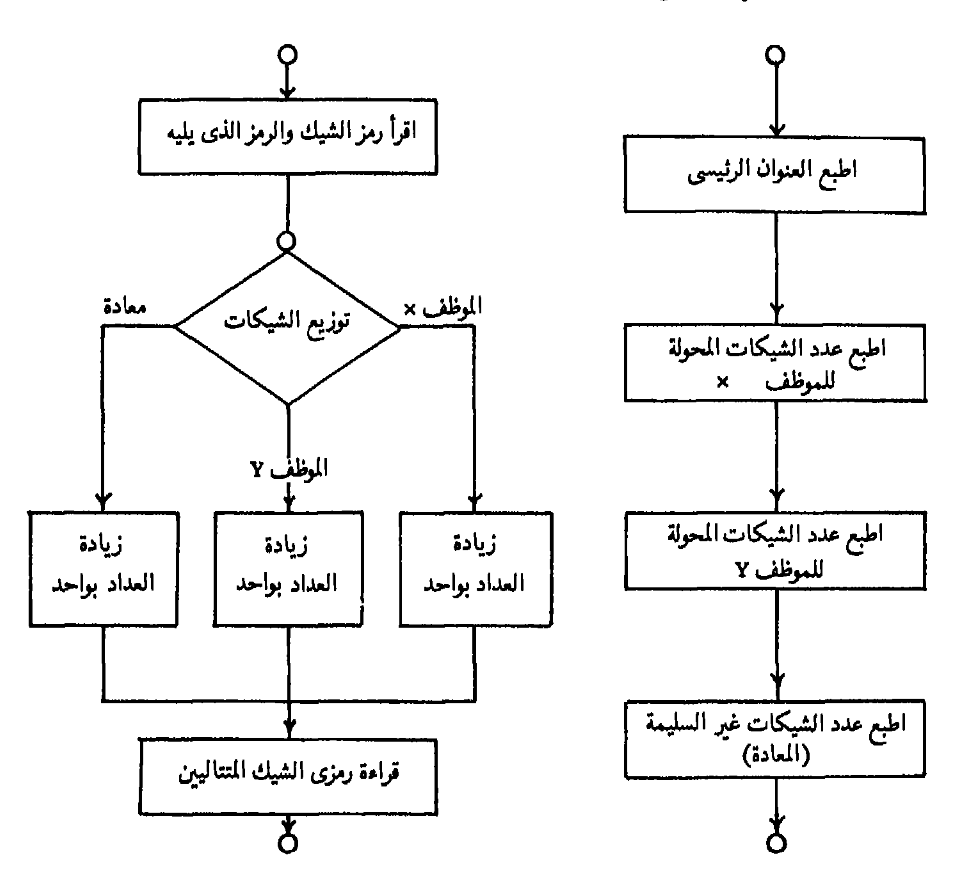
```
IF  A$ < > «A» A    A$ < > «B»    A$ < > «C»     A$ < > «D»
            THEN    N3   =   N3 + 1
```

أى إضافة واحد للعداد N3 فى حالة أن يكون المتغير A$ لا يساوى أياً من A أو B أو C أو D

ثانياً ــ الهيكل الهرمى :



ثالثاً ــ نمط البرمجة التركيبية :



رابعاً ــ البرنامج فى الشكل التالى :

شكل (٨ ــ ٢)

برنامج لإيجاد عدد الشيكات المحولة للموظف x والموظف Y والمعادة

```
10 REM   البرنامج الرئيسي
20 REM  استدعاء برنامج فرعي
30 GOSUB 100
40 REM  استدعاء برنامج فرعي لطباعة النتائج
50 GOSUB 290
60 END
100 REM  برنامج فرعي لحساب عدد الشيكات
110 REM  اسناد القيمة صفر لعدادات الحالا ت الثلا ثه
120 LET N1=0: LET N2 = 0: LET N3 = 0
130 REM  قراءة رمز الشيك و الرمز الذي يليه
135 READ A$,B
140 REM  هل نهاية البيانات
150 IF A$ = "@" THEN ?70
155 REM  x  هل الشيك يخص الموظف
160 IF A$ = "a" OR A$ = "b" OR A$ = "c" THEN N1 = N1 + 1
170 IF A$ = "d" AND B = 14 THEN N2 = N2 + 1
180 REM  y  هل الشيك يخص الموظف
190 IF A$ = "d" AND B <> 14 THEN N1 = N1 + 1
200 REM   هل الشيك غير سليم
210 IF A$ <> "a" AND A$ <> "b" AND A$ <> "c"  AND A$ <> "d" THEN N3 = N3 + 1
220 REM  لقراءة رمز الشيك و الرمز الذي يليه
230 READ A$,B
240 GOTO 140
250 DATA a,12,b,20,c,18,d,5,d,14,a,14,d,14,d,14,a,13
260 DATA b,15,d,14,n,14,r,14,d,14,c,15,d,16,@,0
270 RETURN
290 REM  برنامج فرعي لطباعة النتائج
300 PRINT "================================================================="
310 PRINT "        ";N1;"  x  عدد الشيكات المحوله للموظف"
320 PRINT "        ";N2;"  y  عدد الشيكات المحوله للموظف"
330 PRINT "        ";N3;"     عدد الشيكات المعاده"
340 PRINT "================================================================="
350 RETURN
```

وعند تنفيذ البرنامج ، ستظهر النتائج كالتالى :

```
===============================================
   9    x   عدد الشيكات المحوله للموظف
   5    y   عدد الشيكات المحوله للموظف
   2        عدد الشيكات المعاده
===============================================
```

أساليب تمثيل المقارنات المركبة :

هناك أساليب لتمثيل المقارنات المتعددة بالرسم حتى يسهل فهمها و برمجتها ، مثل جدول القرارات وشجرة القرارات ، كما هوموضح بالرسمين التاليين :

أ) **جدول القرارات :** و ينقسم إلى قسمين ، الأول : تسرد به الاستفسارات وحالاتها (نعم أو لا) ، والثاني : القرارات التي ستؤخذ على ضوء حالات هذه الاستفسارات ، ففي مثالنا يكون الجدول كالتالي :

	الاستفسارات	حالاتها			
IF	رمز الشيك يساوى A أو B أو C	نعم	لا	لا	لا
	رمز الشيك يساوى D	لا	نعم	نعم	لا
	الرقم الذى يلى الرمز يساوى 14		نعم	لا	
THEN	القرارات	حتى يتم تنفيذها			
	يحول للموظف ×	✓		✓	
	يحول للموظف Y		✓		
	غير سليم و يعاد				✓

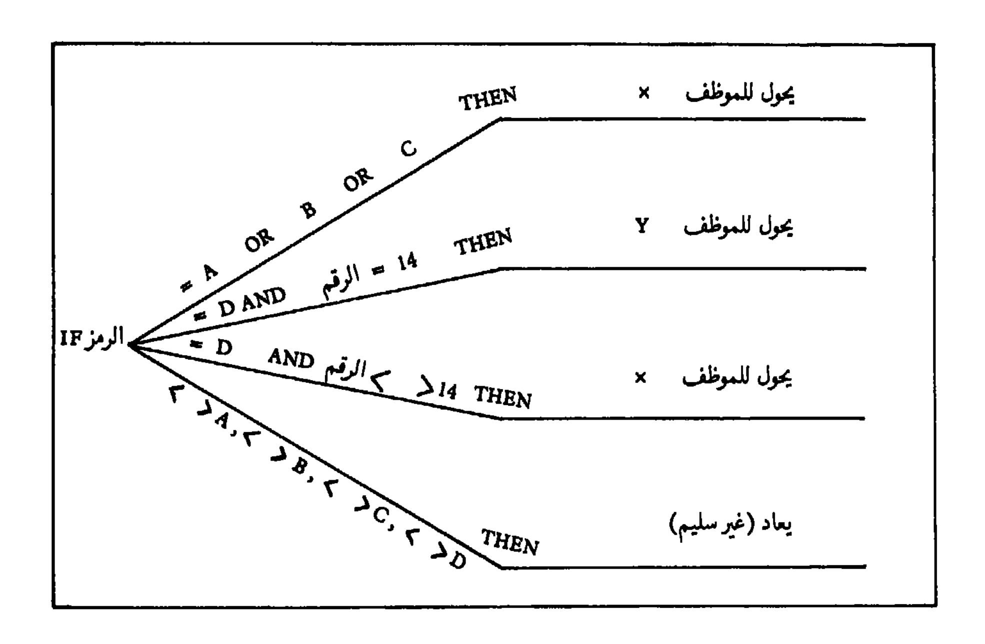
فكما نرى من الجدول :

١ ـ يتم تحويل الشيكات للموظف × في حالتين : أ) إذا كان الرمز يساوى A أو B أو C (نعم) . ب) إذا كان رمز الشيك يساوى D (نعم)، والرقم الذى يلى الرمز لايساوى 14 .

٢ ـ يتم تحويل الشيكات للموظف Y عندما يبين الاستفسار عن رمز الشيك أنه يساوى D (نعم) ، والرقم الذى يلى الرمز يساوى 14 (نعم) .

٣ ـ تتم إعادة الشيك (غير سليم) عندما يبين الاستفسار عن رمز الشيك أنه يساوى A أو B أو C (لا) وعندما يبين الاستفسار عن رمز الشيك أنه يساوى D (لا) .

ب) **شجرة القرار :** وتنقسم فروعها إلى عدد الحالات التى يراد الاستفسار عنها كما هو مبين فى الشكل التالى :

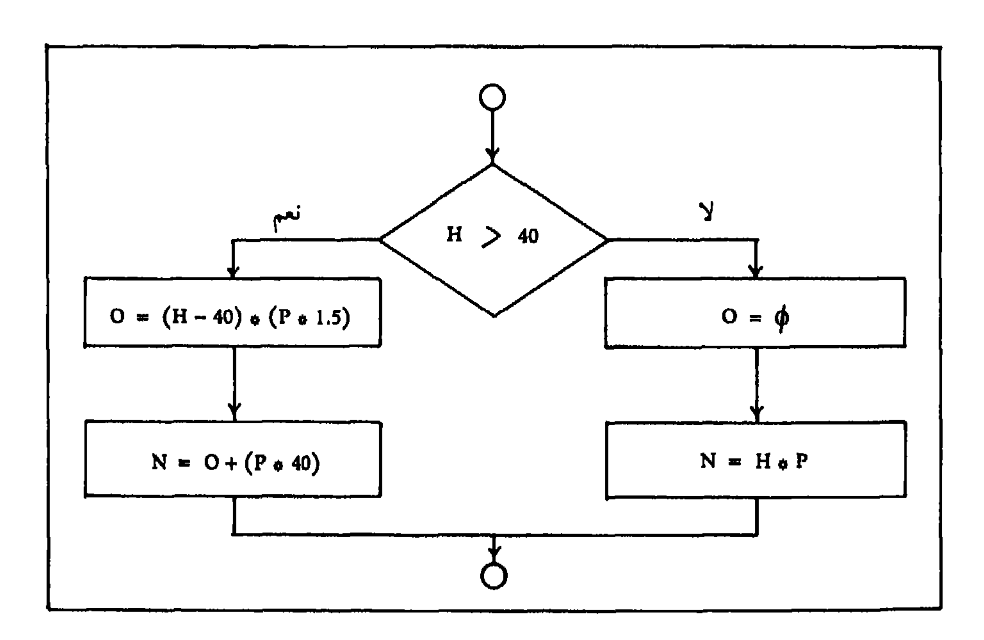


التشعب المتعدد

تطوير برنامج بواسطة تعليمة التشعب المتعدد ON / GOTO

بناء على قيمة معطاة :

تحدثنا فى الجزء السابق عن كيفية التحكم فى سير تنفيذ البرنامج باستخدام تعليمة إذا/إذن IF/THEN ، حيث إن تسلسل تنفيذ البرنامج قد تفرع حسب نتيجة المقارنة بين القيمة التى يحويها المتغير وقيمة ثابتة ، كما هو الحال فى شكل (٨ ـ ١) عبارة (180) ، و يعتبر هذا النوع من التفرع المشروط بذى حدين (مسارين) ، كما هو موضح فى الشكل التالى :



حيث إنه طبقاً للقيمة التى يحتويها المتغير H ، فإن سير تنفيذ البرنامج سيتفرع إما لإيجاد المبلغ المستحق من خارج الدوام ، ومن ثم الراتب الإجمالى فى حالة كون قيمة H أكبر من ٤٠ ساعة أسبوعية ، أو لإيجاد المبلغ الإجمالى دون خارج الدوام فى حالة كون قيمة H تساوى أو أقل من ٤٠ ساعة أسبوعية ، وفى هذه الحالة الأخيرة يكون مبلغ خارج الدوام = صفراً . وتضاف العبارة الأخيرة لجعل المقارنة بين المسارين أكثر وضوحاً وتماثلاً .

ولكن يلاحظ أنه فى بعض الحالات يتطلب تحقيق هدف معين التشعب المتعدد ، بناء على قيمة معطاة لمتغير معين . ومن الأمثلة على هذه الحالة معالجة بيانات تابعة لقطع مختلفة ، أو لمناطق جغرافية متعددة ، أو لآلات إنتاجية ذات وظائف مختلفة ، أو لتبويب البيانات فى أصناف محددة ... وهكذا .

ولتسهيل عملية إجراء مثل هذه المقارنات التى تؤدى إلى التشعب المتعدد ، توجد تعليمة ON/GOTO وسنشرح كيفية استخدام هذه التعليمة فى المثال التالى :

مثال (٨ ـــ ٣) :

الهدف : رجل مبيعات يوجد لديه صنفان من البضاعة ، و يريد معرفة المبلغ الإجمالى لمبيعاته فى نهاية الأسبوع لكلا الصنفين اللذين يبيعهما ، مع العلم بأن كل قطعة من الصنف الأول تباع بمبلغ (١٥٠) ريالاً ، ومن الصنف الثانى تباع بمبلغ (١٧٥) ريالاً .

أولاً ـــ خطوات الحل :

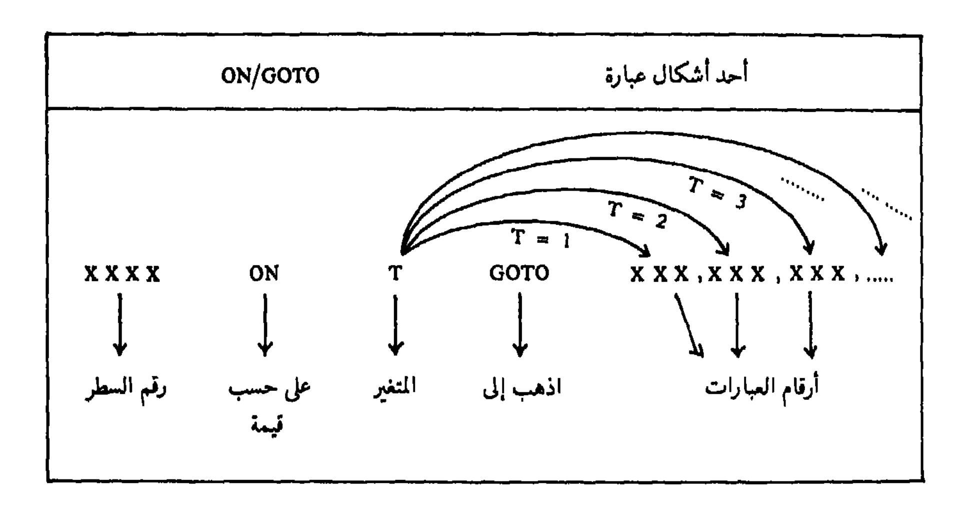
١ ـــ الحصول على عدد القطع المبيعة ورمز كل صنف .

٢ ـــ إذا كان الرمز يساوى ١ نوجد الإجمالى لمبيعات الصنف الأول لذلك اليوم ، ومن ثم نضيفه إلى إجمالى مبيعاته .

٣ ــ إذا كان الرمز يساوى ٢ نوجد الإجمالى لمبيعات الصنف الثانى لذلك اليوم ، ومن ثم نضيفه إلى إجمالى مبيعاته .

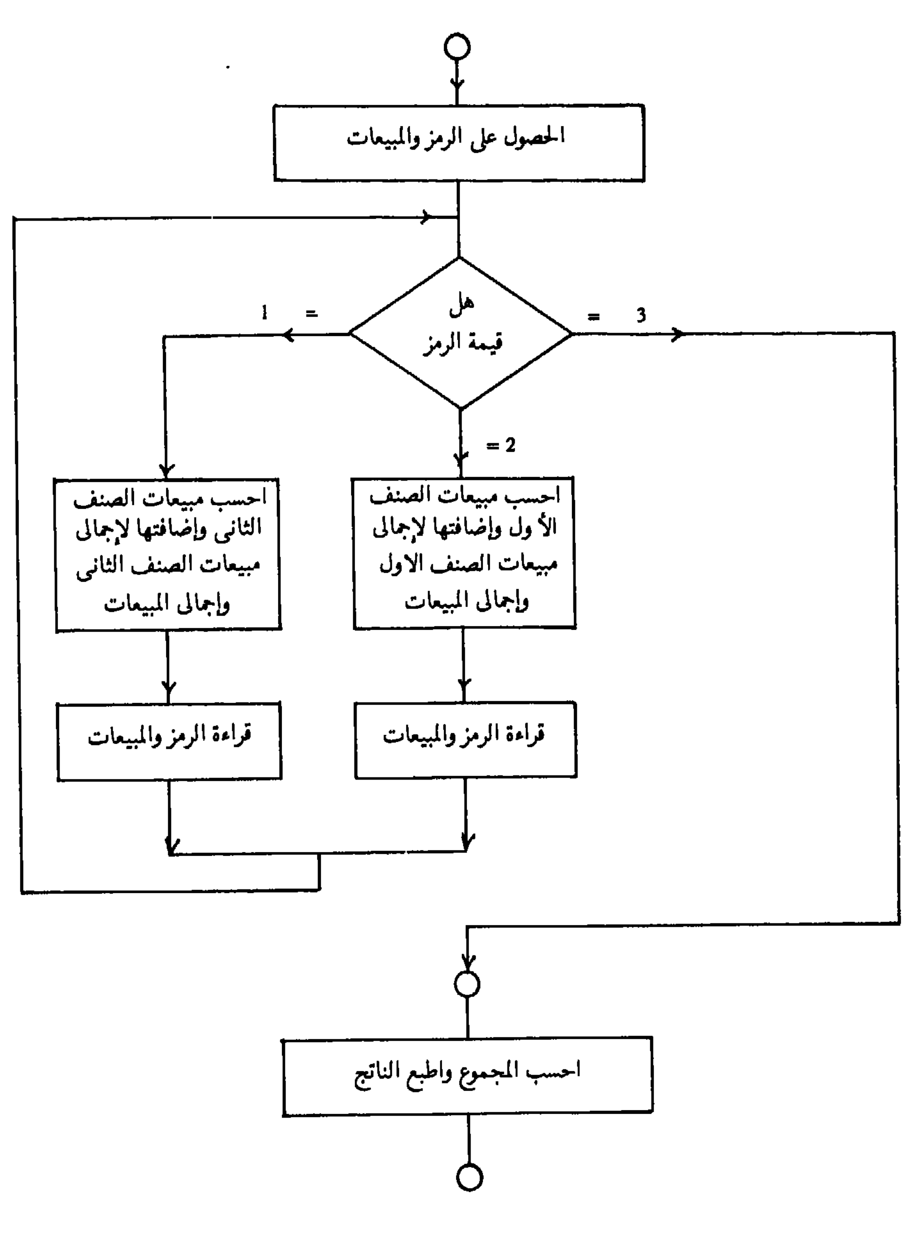
٤ ــ إذا كان الرمز يساوى ٣ نوجد المبلغ الكلى لمبيعات الصنفين ومن ثم طباعة النتائج والتوقف .

نلاحظ من هذا المثال أن القيمة المراد الاستفسار عنها تتراوح مابين ١ ــ ٣ ، وهذا يتطلب منا القيام بأكثر من عملية مقارنة أى تكرار استخدام IF/THEN ، ولكن استخدام تعليمة ON/GOTO سيمكننا من الاستغناء عن ضرورة تكرار عبارة IF/THEN ، وذلك بالتشعب إلى ثلاثة أرقام عبارات مختلفة بناء على صنف المبيعات ، و يكون أحد أشكال هذه التعليمة كالتالى :



و يتم الانتقال إلى أرقام هذه العبارات حسب قيمة المتغير T . كما هو واضح من الأسهم الموصلة بين المتغير T وأرقام العبارات الظاهرة بعد كلمة GOTO ، فإذا كانت القيمة تساوى ١ يتم الذهاب إلى رقم العبارة الأولى الذى يلى GOTO ، وإذا كانت القيمة تساوى ٢ يتم الانتقال إلى رقم العبارة الثانية الذى يلى GOTO ، وهكذا.

ثانياً وثالثاً ــ سنستخدم برنامجاً فرعياً واحداً لهذا المثال . ويمثل الشكل التالى رسم البرمجة التركيبية لهذا الجزء :



رابعاً ــ يحتوى الشكل التالى على البرنامج الكامل لتحقيق الهدف من هذا المثال :

شكل (٨ ــ ٣)

برنامج لايجاد المبلغ الإجمالى فى نهاية الأسبوع لمبيعات صنفين من البضاعة

```
10 REM   استدعاء برنامج فرعي لحساب المبلغ الا جمالي
20 GOSUB 40
30 END
40 REM  اسناد القيمه صفر كقيمه اوليه للمجاميع
50 LET T1= 0
60 LET T2 = 0
70 REM  قراءة الصنف و عدد القطع المباعه
80 READ T,Q
90 REM  الا ستفسار عن نوع الصنف للتشعب في سير تنفيذ العمليات
100 ON T GOTO 110,160,210
110 REM  للا يجاد اجمالي مبيعات اليوم من الصنف الا ول واضافته للمجموع الكلي
120 LET A = Q * 150
130 LET T1 = T1 + A
140 READ T,Q
150 GOTO 100
160 REM  للا يجاد اجمالي مبيعات اليوم من الصنف الثاني واضافته للمجموع الكلي
170 LET A = Q * 175
180 LET T2 = T2 + A
190 READ T,Q
200 GOTO 100
210 REM  للا يجاد المبلغ الا جمالي للمبيعات من الصنفين وطباعة النتائج
220 LET S = T1 + T2
230 PRINT "==================================================================="
240 PRINT " المبلغ الا جمالي لمبيعات الصنف الا ول  =  ";T1;" ريالا سعوديا "
250 PRINT " المبلغ الا جمالي لمبيعات الصنف الثاني  =  ";T2;" ريالا سعوديا "
260 PRINT "==================================================================="
270 PRINT " المبلغ الا جمالي لمبيعات الا سبوع من الصنفين  =  ";S;" ريالا سعوديا "
280 PRINT "==================================================================="
290 DATA 1,30,2,10,1,0,2,5,1,15,2,9,1,8,2,6,1,40,2,21
300 DATA 1,18,2,13,1,7,2,3,3,0
310 RETURN
```

وعند تنفيذ البرنامج ستظهر النتائج كالتالى :

```
======================================================================
 المبلغ الا جمالي لمبيعات الصنف الا ول    =    17700   ريالا   سعوديا
 المبلغ الا جمالي لمبيعات الصنف الثاني    =    11725   ريالا   سعوديا
======================================================================
المبلغ الا جمالي لمبيعات الا سبوع من الصنفين  =   29425   ريالا   سعوديا
======================================================================
```

تطوير برنامج باستخدام تعليمة ON/GOTO
مع إجراء عمليات حسابية للتوصل إلى قيم التفرع :

فى بعض الحالات ستكون القيم التى سيحويها المتغير أكبر بكثير من القيم التى مثلها كل من المتغير T فى شكل (٨ ــ ٣) أو المتغير N فى شكل (٨ ــ ٤) وهذا يوجب علينا وضع أرقام عبارات مساوية للقيمة العليا التى سيحتويها هذا المتغير.

وللتحايل على مثل هذا النوع من القيم يمكننا القيام بعمليات حسابية على هذا المتغير، لجعل هذه القيمة تتناسب مع نوع العمليات المراد إجراؤها، وذلك كما فى المثال التالى :

مثال (٨ ــ ٤) :

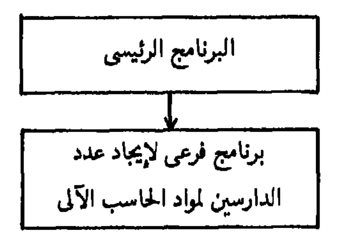
الهدف : قسم الحاسب الآلى فى معهد الإدارة العامة يرغب فى إيجاد عدد الدارسين الذين انضموا لكل من المواد الأربع التالية :

اسم المادة	رمزها
مقدمة فى الحاسب الآلى	١٠١
البرمجة بلغة البيسك	١٠٢
التحليل والتصميم	١٠٥
إدارة المشاريع	١٠٦

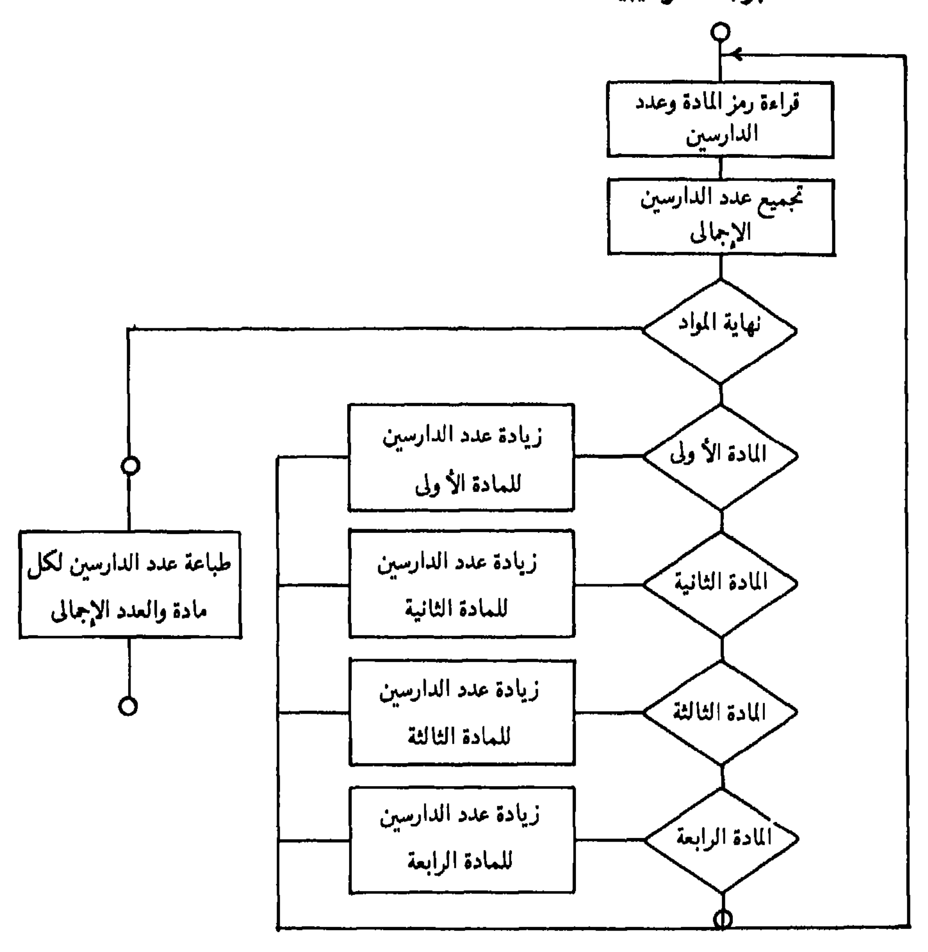
أولاً ــ خطوات الحل :

١ ــ الحصول على رمز المادة وعدد الدارسين .
٢ ــ إضافة عدد الدارسين لمجموع المادة .
٣ ــ إضافة عدد الدارسين للمجموع الكلى للدراسين .

ثانياً : الهيكل الهرمى للبرنامج :



ثالثاً : نمط البرمجة التركيبية :



رابعاً ـ البرنامج كاملاً فى الشكل التالى :

شكل (٨ ـ ٤)

برنامج لإيجاد عدد الدارسين لمواد الحاسب الآلى

```
10 REM        استدعاء برنامج فرعي
20 GOSUB 40
30 END
40 REM  برنامج فرعي لا يجاد عدد الدارسين المنضمين لمواد الحاسب الآ لي
50 REM      قراءة رمز الماده وعدد الدارسين
60 READ N,S
70 REM      للا ستفسار عن نهاية البيانات
80 IF N=0 THEN 240
90 T=T+S
100 ON N-100 GOTO 110,140,60,60,170,200
110 REM  اضافة عدد الدارسين للعدد الا جمالي للمادة الا ولي
120 N1=N1+S
130 GOTO 60
140 REM  اضافة عدد الدارسين للعدد الا جمالي للمادة الثانيه
150 N2=N2+S
160 GOTO 60
170 REM  اضافة عدد الدارسين للعدد الا جمالي للمادة الثالثه
180 N3=N3+S
190 GOTO 60
200 REM  اضافة عدد الدارسين للعدد الا جمالي للمادة الرابعه
210 N4=N4+S
220 GOTO 60
230 DATA 101,5,102,7,103,8,104,6,101,4,102,12,105,12,106,14,0,0
240 REM                                   لطباعة المخرجات
250 PRINT "  عدد الدارسين الذين انضموا الى المواد الا ربع التاليه"
260 PRINT "============================================================"
270 PRINT       TAB(5);"عدد الدارسين";TAB(49);" الماده "
280 PRINT TAB(5);"------------";TAB(49);"------ "
290 PRINT TAB(9);N1;TAB(30);" مقدمه في معالجة البيانات"
300 PRINT TAB(9);N2;TAB(37);" البرمجه بلغة بيسك"
310 PRINT TAB(9);N3;TAB(38);" التحليل والتصميم"
320 PRINT TAB(9);N4;TAB(40);" ادارة المشاريع"
330 PRINT "============================================================"
340 PRINT TAB(9);T;TAB(17);"=  العدد الا جمالي للدارسين"
350 PRINT "============================================================"
360 RETURN
```

وفى حالة تنفيذ البرنامج ، تظهر النتائج كالتالى :

```
        عدد الدارسين الذين انضموا الى المواد الا ربع التاليه
============================================================
    عدد الدارسين                                      الماده
    ------------                                      ------
         9                       مقدمه في معالجة البيانات
         19                               البرمجه بلغة بيسك
         12                                التحليل والتصميم
         14                                 ادارة المشاريع
============================================================
         68     =            العدد الا جمالي للدارسين
============================================================
```

لو تتبعنا البرنامج السابق نجد أن القيم التى سيحتويها المتغير تتراوح مابين ١٠١ إلى ١٠٧ ، وهذا يعنى أننا بحاجة إلى وضع مائة وسبعة أرقام للعبارات تلى GOTO ، لكن يمكننا تقليل ذلك الرقم بإجراء عملية الطرح على هذه القيمة دون أن تؤثر على منطق عمل البرنامج ، وذلك بطرح (١٠٠) من كل قيمة تسند للمتغير، وهذا ماتم فى العبارة (١٥٠) . فالقيمة المستخدمة فى المقارنة للتعبير الجبرى هى ١ ، ٢ ، ٣ ، ٤ ، ٥ ، ٦ ، ٧ وهى مقابلة لرموز المواد ١٠١ ، ١٠٢ ، ١٠٣ ، ١٠٤ ، ١٠٥ ، ١٠٦ ، ١٠٧ (والأخير يعتبر مؤشراً لنهاية البيانات) . كما أن البيانات فى العبارة (٢٨٠) احتوت على رمزى مادتى ١٠٣ ، ١٠٤ وهذان الرمزان خاصان بمادتين أخريين لاعلاقة لهما بما هو مطلوب منا القيام به ، فيتم إهماله لتوجيه الحاسب للقراءة فى كل مرة وجدت بها قيمة التغير تساوى ٣ أو ٤ دون أى تعداد .

وكما أمكننا طرح (100) من قيمة المتغير N يمكننا إضافة قيمة ثابتة للمتغير، فى حالة حصوله على قيم سالبة ، أو ضربه فى قيمة ثابتة ، أو رفعه إلى قوة ثابتة .

أما فى حالة القسمة إذا حصلت كسور عشرية فى القيمة التى ستسند للمتغير فيتم إهمالها ، وسيسند الرقم الصحيح فقط . فمثلا إذا كانت قيمة N (٢٩) وتمت قسمتها على القيمة (١٠) فى المثال التالى :

```
ON    N / 10    GOTO    200 , 250 , 370
```

يتم الذهاب إلى العبارة رقم ٢٥٠ أى تم إهمال (٩,) الكسر العشرى وإسناد القيمة (٢) للمتغير N .

وهذه بعض الأمثلة من عبارة ON/GOTO توجد بها تعبيرات جبرية مختلفة :

```
ON    X - W 10    GOTO    ....
ON    11 * 12 - 100 GOTO    ....
ON    (1 ↑ 2 - J)/20    GOTO    ....
```

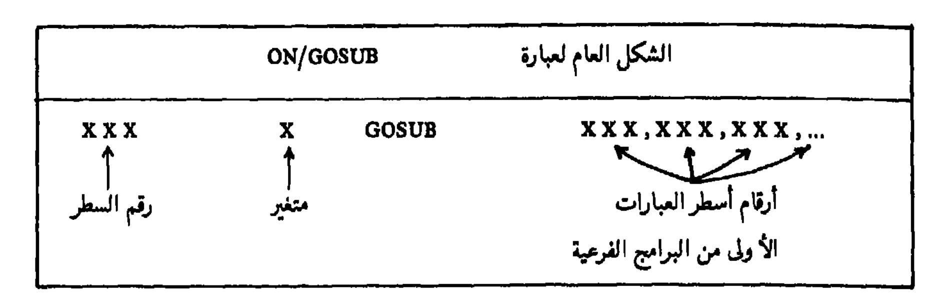
كيفية استخدام تعليمة اذهب إلى
برنامج فرعى والعودة منه ON/GOSUB , RETURN :

إذا تتبعنا خطوات تنفيذ البرنامج شكل (٨ ــ ٣) نجد أن العبارة (٨٠) تعمل على قراءة عدد القطع المبيعة ومن ثم صنف القطع ، وعبارة (100) تستفسر عن نوع الصنف (المتغير T) فإذا كانت قيمة المتغير T تساوى ١ يتم الذهاب إلى عبارة (110) لإيجاد المبلغ الإجمالى لمبيعات اليوم من الصنف الأول ، وإذا كانت قيمة المتغير T تساوى ٢ يتم الذهاب إلى عبارة (160) لإيجاد المبلغ الإجمالى لمبيعات اليوم من الصنف الثانى ، أما إذا كانت قيمة المتغير T تساوى ٣ فيتم الذهاب إلى عبارة (210) لإيجاد المبلغ الإجمالى للمبيعات من الصنفين ومن ثم تتم طباعة النتائج .

وهناك تعليمة أخرى مشابهة لتعليمة ON/GOTO وهى تعليمة ON/GOSUB ، حيث يتم التفرع إلى أرقام العبارات التى تحدد بدايات برامج فرعية بدلاً من أرقام عبارات فى البرنامج الرئيسى ، وذلك بناء على قيمة متغير معين كما فى التعليمة السابقة .

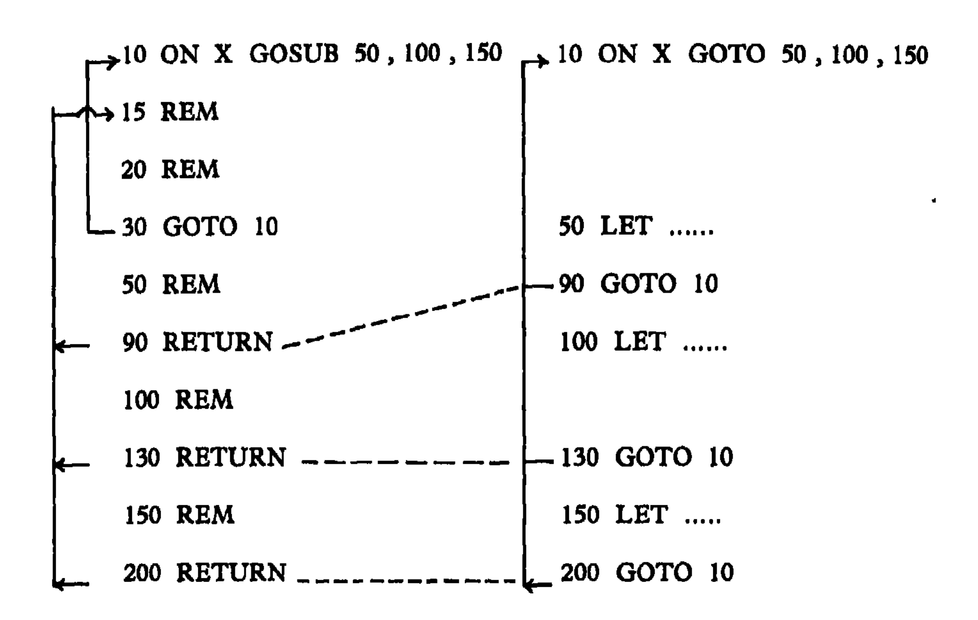
لقد أمكننا التحكم فى تسلسل تنفيذ البرنامج فى شكل (٨ ــ ٣) باستخدام تعليمة ON/GOTO فى العبارة رقم (100) وذلك لتنفيذ أى من العبارات 110 , 160 , 210 اعتماداً على القيمة المخزنة فى المتغير T (١ ، ٢ ، ٣) .

أما إذا استخدمت عبارة ON/GOSUB فإن أرقام العبارات 110 , 160 , 210 ستكون بدايات لبرامج فرعية . وحال الانتهاء من تنفيذ البرنامج الفرعى يتم العودة إلى السطر الذى يلى عبارة ON/GOSUB باستخدام تعليمة RETURN فى كل من البرامج الفرعية . لذلك تستخدم تعليمتا RETURN و ON/GOSUB للتفرع من الأجزاء المتحكمة (البرامج الرئيسية) إلى الأجزاء المنفذة (البرامج الفرعية) والعودة إليها . والشكل العام لهذه العبارة هو كالتالى :



مقارنة تعليمتى ON/GOTO و ON/GOSUB :

و يوضح المثال التالى الفرق بين تعليمتى ON/GOTO و ON/GOSUB



لاحظ الفرق التالى بين التعليمتين : فى حالة ON/GOTO يتم العودة إلى تنفيذ العبارة مرة أخرى بعد التشعب منها عن طريق تعليمة GOTO . أما فى التعليمة الأخرى فيتم العودة إلى السطر الذى يلى ON/GOSUB بواسطة تعليمة RETURN ، ومن ثم تستخدم تعليمة GOTO لتنفيذ عبارة ON/GOSUB مرات أخرى .

ملاحظات على استخدامات تعليمتى ON / GOTO و ON / GOSUB :

١ ـــ يجب أن يكون عدد أرقام أسطر العبارات التى تلى GOTO ، مساو ياً للقيمة العظمى التى سيحو يها المتغير × .

٢ ـــ فى حالة استخدام ON X GOSUB يجب أن يكون عدد أرقام بداية البرامج الفرعية ، التى تلى GOSUB مساو ياً للقيمة العظمى ، التى سيحو يها المتغير × .

٣ ـــ يفضل استخدام التعليمة ON X GOTO أو ON X GOSUB عندما تكون القيمة العظمى للمتغير × معروفة مسبقاً ، وأن تكون القيمة العظمى التى سيحو يها المتغير × لا تزيد على عشرة (١٠) .

٤ ـــ يمكننا استخدام رقم عبارة أو برنامج فرعى لأكثر من قيمة للمتغير × .

مثال أ ـــ ON X GOTO 100 , 150 , 170 , 100 , 200

هنا يتم تنفيذ العبارة (١٠٠) عندما تكون قيمة × تساوى (١) أو (٤) .

مثال ب ـــ ON N GOSUB 100 , 300 , 400 , 100 , 700

هنا يتم استدعاء البرنامج الفرعى الذى يبدأ من العبارة (١٠٠) ، عندما تكون قيم المتغير × تساوى (١ أو ٤) .

٥ ـــ فى حالة حصول المتغير × على قيمة أعلى من عدد أرقام العبارات سواء كانت تابعة لـ GOTO أو GOSUB ، فإن الحاسب إما يعمل على تنفيذ العبارة التالية للعبارة ON × .... ، أو ينهى تنفيذ البرنامج حسب طبيعة الحاسب المستخدم (ارجع للفصل السابع عشر) .

٦ ـــ يمكن استخدام ON/GOTO فى حالات التشعب داخل الجزء الواحد فى الهيكل الهرمى ، وخاصة إذا كانت المعالجات التابعة لكل منها بسيطة . أما فى حالة التشعب لتنفيذ أجزاء متكاملة فى الهيكل الهرمى للبرنامج فيمكن استخدام ON/GOSUB .

تطوير برنامج باستخدام تعليمتى ON/GOSUB و RETURN :

مثال (٨ ــ ٥) :

شركة تأجير سيارات لها ثلاثة مراكز فى المملكة ، الأول فى الرياض ، والثانى بجدة ، والثالث بالدمام ، سياستها فى التأجير أن تسمح للمستأجر أن يقطع يومياً مسافة ١٥٠ كم بالسيارة مقابل مبلغ ١٥٠ ريالاً ، وأن يدفع ٥,٥٠ ريالات عن كل كم يزيد على ذلك.

الهدف : ترغب الشركة فى إيجاد المبلغ الإجمالى للدخل اليومى فى كل من المراكز الثلاثة ، والدخل الإجمالى للشركة .

أولاً ــ خطوات الحل :

١ ــ الحصول على رمز المدينة ، رقم لوحة السيارة ، اسم المستأجر ، عداد السيارة وقت الاستئجار ، عداد السيارة عند الإعادة ، عدد أيام الاستئجار .

٢ ــ إيجاد عدد الكيلومترات المستخدمة .

٣ ــ إيجاد عدد الكيلومترات الزائدة .

٤ ــ إيجاد التكلفة الرئيسية والزائدة .

٥ ــ إيجاد الدخل لكل فرع وللشركة ككل .

٦ ــ طباعة اسم المدينة ، رقم لوحة السيارة ، اسم المستأجر ، عداد السيارة وقت الاستئجار ، عداد السيارة عند الإعادة ، عدد الكيلومترات المصرح بها ، عدد الكيلومترات الزائدة ، التكلفة الرئيسية ، التكلفة الزائدة ، التكلفة الإجمالية لكل فرع ، ومجموع التكلفة الكلية .

ثانياً : يوضح الشكل التالى الهيكل الهرمى ذا المستويات الثلاثة.

ـ الهيكل الهرمى مع أرقام عبارات البرامج فى الشكل (٨ ـ ٥) :

البرنامج الرئيسى

100 إدخال التاريخ

150 قراءة المدخلات

200 الحسابات والطباعة

150 قراءة المدخلات

600 حسابات الفرع الأول والطباعة

340 الحسابات الأولية

620 حسابات الفرع

425 طباعة معلومات الفرع

700 حسابات الفرع الثانى والطباعة

340 الحسابات الأولية

720 حسابات الفرع

425 طباعة معلومات الفرع

800 حسابات الفرع الثالث والطباعة

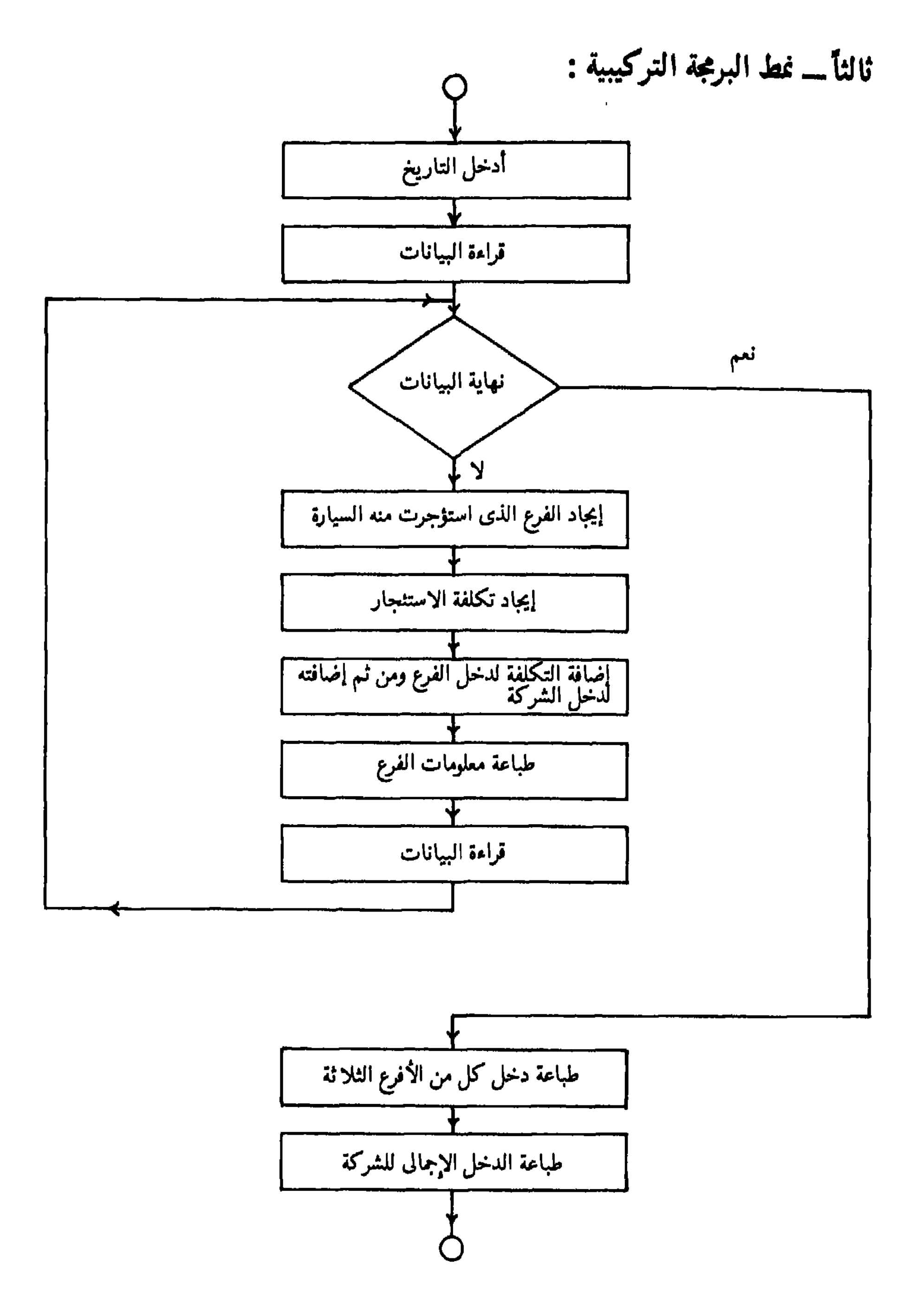
340 الحسابات الأولية

820 حسابات الفرع

425 طباعة معلومات الفرع

900 الطباعة الإجمالية

ثالثاً ـــ نمط البرمجة التركيبية :



رابعاً ــ البرنامج فى الشكل التالى :

شكل (٨ ــ ٥)

برنامج لحساب الدخل الإجمالى لمؤسسة تأجير سيارات والدخل التفصيلى لأفرعها

```
10 REM  استدعاء برنامج فرعي
20 GOSUB 100
30 REM   استدعاء برامج فرعيه
40 GOSUB 300
45 GOSUB 900
47 GOSUB 1000
50 END
100 REM  برنامج فرعي للحصول على التاريخ وطباعة العناوين
120 INPUT "00,00,0000  ادخل التاريخ اليوم  الشهر السنه كالتالي";D,M,Y
140 PRINT "====================================================================
========"
150 PRINT TAB(20);" شركة ا ب ج  لتاجير السيارات ";TAB(60);Y;"/";M;"/";D
155 PRINT "====================================================================
========"
160 PRINT " رقم     اسم     العداد  عند  عدد  كم   كم     تكلفه    تكلفه
  اسم"
170 PRINT " اللوحه    الزبون  استئجار اعاده الايام مستخدم زائد    زائده    كليه
  المدينه"
190 PRINT "====================================================================
========"
200 RETURN
300 REM   برنامج فرعي للحصول على البيانات
310 REM
320 READ N,P$,N$,M1,M2,D
325 REM  للا ستفسار عن نهاية البيانات
327 IF N = 0 THEN RETURN
330 REM   للحصول على عدد الكيلومترات المستخدمه
340 LET M = M2 - M1
350 REM   للا يجاد  عدد الكيلومترات المسموحه
360 LET A = D * 150
370 REM   للا يجاد عدد الكيلومترات الزائده
375 LET K = M - A
380 REM   للا يجاد تكلفة الكيلومترات  الزائده
385 LET E = K * 3.5
386 REM   للا يجاد التكلفه الكليه
390 LET C = E  + D * 100
420 ON N GOSUB 600,700,800
425 PRINT TAB(1);C;TAB(10);E;TAB(19);K;TAB(27);M;TAB(34);D;TAB(37);M2;TAB(44);M1
;TAB(52);N$;TAB(65);P$;C$
430 REM
440 GOTO 320
450 REM      البيانات
500 DATA 1,24561,ALI AHMED,25430,28340,7
510 DATA 2,31890,HASAN SALIM,20710,34560,2
520 DATA 3,72098,RADI AHMED,45430,48340,5
530 DATA 2,90811,ALI HASAN,55430,58040,9
540 DATA 1,56901,ANAS ALI,44567,46098,6
550 DATA 3,10981,SAMEER RADI,8123,12450,8
580 DATA 0,00000,AAAAAAAA,00000,00000,0
600 REM  برنامج فرعي للا يجاد دخل فرع الرياض  واضافته على الدخل الكلي
620 LET C$ = " الرياض "
630 LET R1 = R1 + C
640 T = T + C
650 RETURN
700 REM  برنامج فرعي للا يجاد دخل فرع  الدمام  واضافته على الدخل الكلي
710 LET C$ = "  الدمام "
720 LET R2 = R2 + C
730 LET T = T + C
740 RETURN
```

```
800 REM  برنامج فرعي للا يجاد دخل فرع جده  واضافته على الدخل الكلي
810 LET C$ = "    جده  "
820 LET R3 = R3 + C
830 LET T = T + C
840 RETURN
900 REM  برنامج فرعي  لطباعة الدخل الا جمالي لكل فرع
910 PRINT "=====================================================================
========"
920 PRINT TAB(20);R1;TAB(30);" =  اجمالي دخل فرع  الرياض"
930 PRINT TAB(20);R2;TAB(30);" =  اجمالي دخل فرع الدمام"
940 PRINT TAB(20);R3;TAB(30);" =  اجمالي دخل فرع جده"
945 PRINT "=====================================================================
========"
950 RETURN
1000 REM  برنامج فرعي للا يجاد الدخل الا جمالي للشركه و طباعته
1020 LET R = R1 + R2 + R3
1030 PRINT TAB(20);R;TAB(30);" =  اجمالي الدخل للشركه"
1035 PRINT "=====================================================================
========"
1040 RETURN
1105 PRINT
```

وفى حالة تنفيذ البرنامج السابق سنحصل على النتائج التالية :

شركة ا ب ج لتاجير السيارات — 1987 / 12 / 30

اسم المدينه	رقم اللوحه	اسم الزبون	العداد عند استئجار	عند اعاده	عدد الايام	كم مستخدم	كم زائد	تكلفه زائده	تكلفه كليه
الرياض	24561	ALI AHMED	25430	28340	7	2910	1860	6510	7210
الدمام	31890	HASAN SALIM	20710	34560	2	13850	13550	47425	47625
جده	72098	RADI AHMED	45430	48340	5	2910	2160	7560	8060
الدمام	908[illegible]	ALI HASAN	55430	58040	9	2610	1260	4410	5310
الرياض	56901	ANAS ALI	44567	46098	6	1531	631	2208.5	2808.5
جده	10981	SAMEER RADI	8123	12450	8	4327	3127	10944.5	11744.5

10018.5 = اجمالي دخل فرع الرياض

52935 = اجمالي دخل فرع الدمام

19804.5 = اجمالي دخل فرع جده

82758 = اجمالي الدخل للشركه

تمارين

١ ـ اعمل على طباعة وتنفيذ البرامج الموجودة فى أشكال الفصل، وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعليمات .

٢ ـ بيّن الأخطاء إن وجدت فى كل من العبارات التالية :

أ
```
10      IF A = 5 THEN B = 2
```
ب
```
10      IF A = 4 THEN 70 ELSE B = 12
```
جـ
```
10      IF A = 4 AND 7 THEN 80
```
د
```
10      IF A > A THEN B = A
```
هـ
```
10      IF B < 10 THEN ELSE C = B
```

٣ ـ بين نتائج تنفيذ كل من البرامج التالية :

أ ـ
```
10      READ A , B
20      IF A > B THEN C = A ELSE C = B
30      PRINT C , A , B
40      DATA 10 , 7 , 2 , 4
50      END
```

ب ـ
```
10      LET I = 0
20      READ A
30      IF I > 5 THEN 90
40      IF A > 0 AND A < 10 THEN I = I + 1
50      PRINT I , A
60      READ A
```

```
70        COTO  20
80        DATA  10 , 20 , 7 , 2 , 1 , 3 , 4 , 17
90        END
```

٤ ــ بيّن القيم التى يجب أن يحتويها المتغير A لكى يتم تنفيذ العبارات التالية بدون أخطاء .

أ
```
10    ON  A     COTO              10 , 20 , 30 , 40
```
ب
```
10    ON  A – 10  COTO            70 , 80 , 90 , 100 , 90
```
جـ
```
10    ON  ((A / 100) – 1)  COTO   60 , 50 , 70 , 90
```
د
```
10    ON  A * .5  COSUB           100 , 200 , 300
```

٥ ــ اكتب برنامجاً يعمل على قراءة بيانات أربعة أصناف من البضائع ، مع العلم بأن بيانات الصنف مكونة من (رقم الصنف ، عدد القطع المبيعة ، سعر كل قطعة) ومن ثم إيجاد الثمن الإجمالى لمبيعات كل صنف .

أ ــ باستخدام تعليمة IF / THEN

ب ــ باستخدام تعليمة ON / GOTO

٦ ــ محل السلمان لبيع أدوات وملحقات التصوير يمنح خصماً لزبائن الجملة كالتالى :

تكلفة البضائع المشتراة	الخصم ٪
١ ــ أقل من ٢٠٠ ريال	صفر
٢ ــ من ٢٠٠ وإلى أقل من ٥٠٠ ريال	٥
٣ ــ من ٥٠٠ وإلى أقل من ٨٠٠ ريال	٧
٤ ــ ٨٠٠ ريال فأكثر	١٠

طور برنامجاً لعمل التالى :

١ . أدخل أرقام القطع وأسعارها وكمياتها لكل زبون .

٢ . احسب المجموع .

٣ . احسب الخصم .

٤ . احسب صافى المبلغ المستحق من زبون الجملة .

٧ ــ محل لبيع الأسماك بالجملة يستقبل طلبيات من أنواع ثلاثة من الزبائن :

ــ مطاعم .

ــ مجمعات سكنية .

ــ خاص من أفراد .

ومن الممكن أن تحتوى الطلبية على الأصناف التالية : السمك ، الخبز ، والحلويات . وتوجد تسعيرة خاصة لكل نوع من الزبائن مع خصم من مجموع قيمة الطلبية كالتالى :

١٠٪ للمطاعم فوق ٢٠٠ ريال ، ٧٪ للسكنى فوق ١٥٠ ريالاً ، ٣٪ للأفراد فوق ٣٠ ريالاً .

طور برنامجاً لحساب قيمة الفاتورة لكل طلبية ، وإيجاد مجموع المبيعات لكل نوع من الزبائن والمجموع الكلى للمبيعات فى نهاية كل يوم .

٨ ــ يقوم مكتب محامين بتحصيل الديون المتأخرة للشركات ، و يقوم المكتب بحساب العمولة كنسبة من المبلغ المحصل ، وذلك حسب الجدول التالى :

المبلغ المحصل بالريال	العمولة ٪
أقل من ٢٠٠٠	٣
٢٠٠٠ إلى أقل من ٤٠٠٠	٧
٤٠٠٠ إلى أقل من ٧٠٠٠	١٠
٧٠٠٠ فأكثر	١٥

٩ ــ يؤجر محل (س ص ع) أشرطة الفيديو. ويمثل الجدول التالى أجر اليوم لأنواع الأشرطة المختلفة .

نوع الشريط	الأجر لليوم (بالريال)
١ ــ محاضرات	١٥
٢ ــ أفلام كرتون	٢٠
٣ ــ أفلام عادية للأطفال	٣٠
٤ ــ أفلام عادية للكبار	٣٥

يود صاحب المحل الحصول على مجموع الإيرادات لكل نوع من الأشرطة والمجموع الكلى للإيرادات ، إذا علم عدد الأشرطة المؤجرة وعدد الأ يام .

١٠ ــ يحتوى أحد المجمعات السكنية على أربعة أنواع من الشقق ، ولكل نوع منها إيجار شهرى ، كما هو مبين فى الجدول التالى :

نوع الشقة	الإيجار (بالريال)
شقة أ غرفة نوم واحدة	٨٠٠
شقة ب غرفتا نوم	١٠٠٠
شقة جـ ٣ غرف نوم	١٢٥٠
شقة د ٤ غرف نوم	١٥٠٠

بالإضافة إلى الإيجار يقوم صاحب المجمع بتوزيع مصاريف الإنارة الخارجية والمياه بنسبة ١ : ٢ : ٣ : ٤ وذلك حسب نوع الشقة .
طور برنامجاً لحساب مجموع الإيجار الشهرى للشقة الواحدة .

الفصل التاسع

استخدام الدالا ت
فى العمليات الحسابية

فى كثير من التطبيقات ، نجد أن أكثر من برنامج يريد القيام بنفس المعالجة أو أن البرنامج الواحد يكرر تنفيذ نفس المعالجة . وقد أدى هذا إلى استخدام البرامج الفرعية كما أشير إليه فى الأمثلة السابقة ، حيث يتم تنفيذ التعليمات المبينة فى البرنامج الفرعى بواسطة تعليمة COSUB .

ومن هذه العمليات الحسابية المتكررة الاستخدام مايمكن تمثيله بمعادلة جبرية أو أكثر . وهذه المعادلة قد تكون عامة ، يمكن استخدامها فى كل التطبيقات ، أو خاصة بتطبيق معين ، ومثال النوع الأول حساب جيب زاوية معينة ، والنوع الثانى تحويل درجة حرارة من مئوية إلى فهرنهايتية . وقد احتوت لغة بيسك تعليمات خاصة للتعامل مع المعادلات من هذا النوع ، وذلك عن طريق استخدام أسماء الدالات التى ستقوم بالعمليات الحسابية . ففى النوع الأول دالات مبرمجة ومخزنة فى الحاسب وذات أسماء تحدد طبيعة العملية الحسابية التى ستقوم بها ، مثل : جيب ، القيمة المطلقة ... إلخ .

أما فى النوع الثانى فتسمح لغة بيسك للمبرمج بأن يحدد اسماً للدالات والمتغيرات المتضمنة بها والعلاقة الجبرية فيما بينها . و يتم استخدام عبارات الدالات هذه فى البرنامج بحرية تامة من قبل المبرمج وفى أى موضع يشاء، وسنتحدث فى هذا الفصل عن كيفية استخدام هذه الدالات .

الدالات المبرمجة / المبنية Built - in

مقدمة عن الدالات المبرمجة :

إذا أردنا إيجاد جيب الزاوية (أ جـ ب) SINE فى المثلث القائم الزاوية ، نجد

أنها تساوى $= \frac{\text{طول الضلع أ ب}}{\text{طول الضلع أ جـ}}$

أ

ب

جـ

س

هذا فى حالة معرفة أطوال أضلاع المثلث .

أما فى حالة عدم معرفة أطوال أضلاع المثلث ومعرفة الزاوية (س) نجد أن جيب الزاوية (س) =

$$\text{س} - \left(\frac{\text{س}^{٣}}{\text{مفكوك ٣}} + \frac{\text{س}^{٥}}{\text{مفكوك ٥}} + \frac{\text{س}^{٧}}{\text{مفكوك ٧}} \right) + \left(\frac{\text{س}^{٩}}{\text{مفكوك ٩}} - \frac{\text{س}^{١١}}{\text{مفكوك ١١}} \right)$$

$$+ \left(\frac{\text{س}^{١٣}}{\text{مفكوك ١٣}} - \frac{\text{س}^{١٥}}{\text{مفكوك ١٥}} \right) + (\dots\dots) \dots\dots$$

و يتضمن إيجاد جيب الزاوية س° حسب المعادلة السابقة، الخطوات التالية :

١ ــ إيجاد قيم البسط حيث إن قيمة س مرفوعة إلى قوة الأرقام الفردية ٣ ، ٥ ....

٢ ــ إيجاد مفكوك كل من هذه الأرقام الفردية (أى إيجاد عناصر كل رقم منها وضربها بعضها فى بعض) ومن ثم إيجاد الناتج .

٣ ــ قسمة ناتج الخطوة الأولى على ناتج الخطوة الثانية وإضافة خارج القسمة إلى الناتج الكلى .

٤ ــ تكرار الخطوات ١ ــ ٣ بعدد المرات اللازمة لتحقيق الدقة المطلوبة فى الإجابة .

٥ ــ طرح الناتج الكلى من قيمة الزاوية س حيث إن ناتج الطرح يساوى جيب الزاوية .

فلو أراد عدة مبرمجين الحصول على جيب الزاو ية ، كل فى تطبيقه الخاص به ، لاضطر كل واحد منهم لكتابة برنامج فرعى بشكل خاص ومستقل عن الآخر ، وكذلك يمكن لأى منهم أن يستعير البرنامج الفرعى من برنامج آخر وطباعته فى برنامجه ، أو أن تخزن الجداول الخاصة بجيب الزاو ية ــ وغيرها من حساب المثلثات ــ فى البرنامج وتسترجع عند الحاجة إليها .

وأى من هذه الإجراءات يبدو مستهلكاً للوقت وذا فاعلية قليلة ، لذلك فهذه وغيرها من العمليات الشبيهة بها قد تمت برمجتها كما مر ذكره ، بحيث يتمكن المبرمج من استدعاء الدالة المطلوبة باستخدام الاسم الخاص بها فقط مصحوباً بالرقم المناسب لكل دالة ، ومقدار الزاو ية فى هذه الحالة .

وتتبع قواعد الدالات الشكل العام التالى :

الشكل العام لعبارة الدالة FUNCTION			
(Y Y Y)	X X X	X X X X	X X X X X
↓	↓	↓	↓
متغير/قيمة/ عملية حسابية	اسم الدالة	تعليمة	رقم السطر

ففى مثالنا للحصول على جيب الزاو ية (SINE) نستخدم العبارة التالية :

```
10        PRINT        SIN    (S)
```

فإذا كانت قيمة S = 30 نجد أن القيمة المطلوبة ستكون 0.5*

بعض الدالات المبرجة والمزودة فى معظم الحاسبات :

١ ــ للحصول على جيب الزاو ية نستخدم الدالة SIN

مثال : جيب الزاو ية 20

```
10    LET        A = SIN    (20)
```

٢ ــ للحصول على جيب تمام الزاوية نستخدم الدالة COS

مثال : جيب تمام الزاوية 40

```
10    LET    A = COS    (40)
```

٣ ــ للحصول على ظل الزاوية نستخدم الدالة TAN

مثال : ظل الزاوية 40

```
10    LET    A = TAN    (40)
```

٤ ــ للحصول على الجذر التربيعى نستخدم الدالة SQR

مثال : أوجد الجذر التربيعى. نستخدم الدالة 70

الحل :

```
10    LET    A = SQR    (70)
```

٥ ــ للحصول على قيمة الزاوية فى حالة معرفة ظلها نستخدم الدالة ANT

مثال : أوجد قيمة الزاوية التى ظلها = (واحداً)

الحل :

```
10    PRINT    ATN    (1)
```

وفى حالة تنفيذ هذه العباره نجد أن القيمة المطبوعة = ٩٥

٦ ــ للحصول على اللوغاريتم الطبيعى نستخدم الدالة LOG

مثال : أوجد اللوغاريتم الطبيعى للقيمة 100

الحل :

```
10    LET    A = LOG    (100)
```

٧ ــ للحصول على القيم صحيحة بدون كسور نستخدم الدالة INT ، حيث إنها تعطينا القيم الصحيحة وتهمل الكسور فى حالة كون القيم موجبة ، أما فى حالة كون القيم سالبة فإنها تعمل على تقريبها للعدد الصحيح .

مثال :

```
10    LET    A = 15.94
20    LET    B = -24.31
```

```
30    PRINT    INT (A) , INT(B)
40    END
```

وفى حالة تنفيذ هذه العبارات نجد أن القيم المطبوعة هى :

15 -25

أى تم تقريب القيمة السالبة إلى العدد الصحيح ، وأهملت الكسور فى حالة القيمة الموجبة .

٨ ـــ للحصول على القيمة المطلقة ، أى تحويل القيمة من سالبة إلى موجبة ، نستخدم الدالة . ABS

مثال : أوجد القيمة المطلقة للقيم 15 . 94 , 31 . 24-

الحل :

```
10    LET    A = 15 . 94
20    LET    B = 24 . 31
30    PRINT    ABS (A) ,    ABS (B)
40    END
```

وفى حالة تنفيذ هذه العبارة نجد أن القيم المطبوعة هى :

15 . 94 24 . 31

٩ ـــ للحصول على الدالة الأسية نستخدم الدالة EXP وهذا يعنى أن القيمة المراد إيجاد الدالة الأسية لها ستكون (القوة/الأس) للقيمة 2.718282.

مثال : أوجد الدالة الأسية للقيمة 2

الحل :

```
10    LET    A = 2
20    PRINT  EXP (A)
30    END
```

وفى حالة تنفيذ هذه العبارة ستكون النتيجة = 7.389057

١٠ ـ للحصول على النسبة التقريبية نستخدم الدالة PI

مثال : أوجد النسبة التقريبية

الحل :

```
10    PRINT    PI
```

وفى حالة تنفيذ هذه العبارة ستكون النتيجة المطبوعة = 3.1415926

١١ ـ للحصول على إشارة القيمة نستخدم الدالة SGN ، ففى حالة كون النتيجة سالبة ستكون الإشارة تساوى 1- ، وفى حالة كون النتيجة صفراً ستكون الإشارة = 0 أما فى حالة كون النتيجة موجبة فستكون الإشارة = 1+

مثال : أوجد إشارة كل من القيمة 112 و 0 و 30

الحل :

```
10    PRINT SGN (112) , SGN (0) , SGN (-30)
```

وفى حالة تنفيذ هذه العبارة ستكون النتائج المطبوعة كما يلى :

```
+1    0    -1
```

١٢ ـ للحصول على الباقى بعد إجراء العمليات الحسابية نستخدم الدالة MOD

مثال : أوجد باقى قسمة A على B, مع العلم بأن 2 = B, 5 = A

الحل :

```
10    LET    A = 5
20    LET    B = 2
30    PRINT    A    MOD B
40    END
```

وفى حالة تنفيذ هذه العبارة ستكون النتيجة المطبوعة = 0.5

١٣ ـ للحصول على الأرقام العشوائية (RANDOM NUMBERS) نستخدم الدالة RND ، وتستخدم هذه الدالة فى حالة عدم توفر البيانات الحقيقية المراد استخدامها فى المعالجة ؛ لذا يتم استخدام هذه الدالة لتزويدنا بقيم تقريبية .

<table>
<tr><th colspan="4">والشكل العام لعبارة الدالة RND</th></tr>
<tr><td>أ : RND
اسم الدالة</td><td>X X X
رقم السطر</td><td>ب : (X) RND
قيمة أو اسم الدالة
اسم متغير على أن تكون
قيمته بدون كسر .</td><td>X X X
رقم السطر</td></tr>
</table>

وباستخدام الدالة RND سنحصل على أرقام تتراوح قيمتها مابين :
(.000000 - .1.0) .

تطوير برامج باستخدام الدالات المبرمجة :

وبعد هذا العرض الإجمالي للدالات ، سنتعرض لتطوير برامج باستخدام البعض منها .

مثال (٩ ــ ١) :

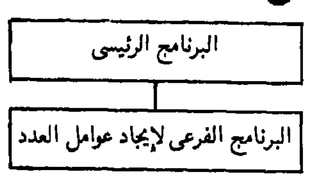
الهدف : إيجاد عوامل (FACTORS) عدد معين ، مع العلم بأن عوامل أى عدد هى العناصر التى يتكون منها العدد بحيث تكون مساوية له أو أقل منه ، ويقبل قسمة العدد عليها بدون كسر (أى بدون بواقٍ) ، فمثلاً عوامل الرقم خمسة هى خمسة وواحد .

أولاً ــ خطوات الحل :

١ ــ الحصول على العدد .
٢ ــ إيجاد عنصر العدد الذى سيكون مساوياً (١ إلى قيمة العدد)
٣ ــ إيجاد ناتج قسمة العدد على العنصر .
٤ ــ الاستفسار عما إذا كان هذا العنصر عاملاً نعمل على طباعته ، وغير ذلك يهمل .
٥ ــ نكرر الخطوات ٢ ــ ٤ حتى نصل إلى قيمة العدد .

ويمكننا عمل ذلك باستخدام إحدى الدالتين INT أو MOD .

ثانياً ــ الهيكل الهرمى للبرنامج :



ثالثاً ــ نمط البرمجة التركيبية :

الحصول على العدد
هل العدد = صفراً
نعم
لا
اجعل B تساوى صفراً
هل قيمة B أكبر من A
نعم
لا
أوجد ناتج قسمة A على B
هل ناتج القسمة كسر
نعم
لا
اطبع قيمة B
أضف واحداً إلى قيمة B

رابعاً ـ أ . البرنامج فى الشكل التالى ، باستخدام دالة INT :

شكل (٩ ـ ١)

برنامج لإيجاد عوامل عدد باستخدام الدالة INT

```
10 REM استدعاء برنامج فرعي
20 GOSUB 40
30 END
40 REM برنامج فرعي للا يجاد عوامل عدد
50 INPUT " ادخل العدد المراد آيجاد عوامله أو صفر 0 للتوقف ";A
60 IF A = 0 THEN 200
70 REM اسناد القيمه الا وليه للمتغير B الذي سيحتوي على عناصر العدد
80 LET B = 0
90 REM للا ستفسار عن اليمة العنصر اذا كانت اكبر من قيمة العدد
100 IF B > A THEN 180
110 REM آيجاد ناتج قسمة العدد على العنصر بدون كسور
120 LET C = INT(A/B)
130 REM للا ستفسار اذا كان العنصر عامل
140 IF (C*B) <> A THEN 160
145 PRINT TAB(10);A;TAB(13);" احدى عوامل العدد ";TAB(33);B;TAB(39);" الرقم "
150 REM اضافة واحد لمتغير العناصر و العوده للا ستفسار
160 LET B = B + 1
170 GOTO 100
180 PRINT "_______________________________________________________"
190 GOTO 50
200 RETURN
```

ب ـ تعديل البرنامج السابق لإيجاد عوامل العدد باستخدام دالة MOD حيث نعمل على تغيير كل من العبارات التالية :

(١) عبارة رقم 120

120 LET C = A MOD B

وهنا تتم قسمة قيمة المتغير A على قيمة المتغير B وسيحتوى المتغير C على باقى القسمة (الكسور) . ومن ثم نستفسر عن قيمة المتغير C . فإذا كانت القيمة لا تساوى صفراً (0) فذلك يعنى أن قيمة المتغير B لا تعتبر عاملاً من عوامل العدد . و يتم ذلك بتغيير العبارة رقم (140) إلى :

140 IF C < > O THEN 180

وبعد تنفيذ البرنامج حسب الطريقتين ، سنحصل على نفس النتائج كالتالى :

الرقم	1	أحد عوامل العدد	6
الرقم	2	أحد عوامل العدد	6
الرقم	3	أحد عوامل العدد	6
الرقم	6	أحد عوامل العدد	6

الرقم	1	أحد عوامل العدد	7
الرقم	7	أحد عوامل العدد	7

الرقم	1	أحد عوامل العدد	9
الرقم	3	أحد عوامل العدد	9
الرقم	9	أحد عوامل العدد	9

مثال (٩ ــ ٢) :

الهدف : الحصول على عشرة أرقام عشوائية* .

أولاً ــ خطوات الحل :

١ ــ إسناد القيمة الأولية 1 لعداد الأرقام العشوائية .

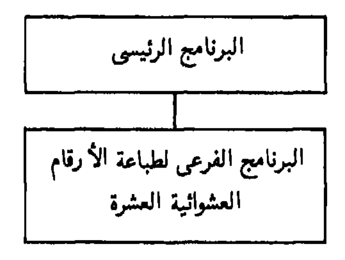
٢ ــ طباعة الرقم العشوائى .

٣ ــ زيادة العداد بواحد والاستفسار عن عدد الأرقام .

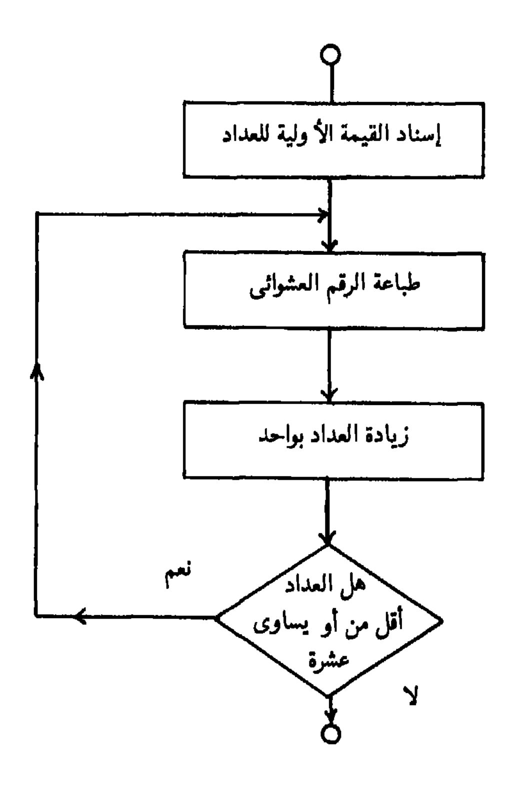
٤ ــ إذا كان عدد الأرقام المطبوعة أقل من عشرة نكرر الخطوات ٢ ــ ٣ وإلا نتوقف .

* إن اختيار رقم عشوائى من بين عدة أرقام ، يعنى أن أى رقم منها له نفس الاحتمال أو الفرصة فى أن يتم اختياره .

ثانياً ــ الهيكل الهرمى للبرنامج :



ثالثاً ــ نمط البرمجة التركيبية :



رابعاً ــ البرنامج فى الشكل التالى :

شكل (٩ ــ ٢)

برنامج للحصول على عشرة أرقام عشوائية باستخدام الدالة RND

```
10 REM          استدعاء برنامج فرعي
20 GOSUB 40
30 END
40 REM  برنامج فرعي للحصول على عشرة ارقام عشوائية
50 REM    اسناد القيمة الاولية (١) لعداد الارقام
60 LET I = 1
70 REM          طباعة الرقم العشوائي
80 PRINT RND
90 REM          زيادة عدد الارقام بواحد
100 LET I = I + 1
110 REM للاستفسار عن عداد الارقام اذا كان اقل او يساوي عشرة يذهب للطباعة
120 IF I <= 10 THEN 80
130 REM         العودة للبرنامج الرئيسي
140 RETURN
```

وفى حالة تنفيذ البرنامج ، ستظهر النتائج كالتالى :

```
.1213501
.651861
.8688612
.7297625
.7988531
7.369809E-02
.4903128
.454519
.1072496
.9505102
```

ولو أردنا الحصول على عشرة أرقام عشوائية أخرى ، وقمنا بتنفيذ البرنامج شكل (٩ ــ ٢) ، لوجدنا أن النتائج (القيمة التى ستطبع) ستكون نفس القيم التى حصلنا عليها سابقاً . وللحيلولة دون ذلك نستخدم الدالة RANDOMIZE فى بداية البرنامج كما هومبين فى الشكل التالى :

الشكل العام لعبارة الدالة	RANDOMIZE*
RANDOMIZE	x x x
اسم الدالة	رقم السطر

* جهاز IBM يتطلب طباعة رقم بين 32768 إلى 32767
بعد اسم الدالة أو عند التنفيذ

فإذا أضفنا العبارة التالية فى بداية البرنامج الفرعى فى شكل (٩ ــ ١) ، نجد أنه فى كل مرة يتم بها تنفيذ البرنامج ، سنحصل على عشرة أرقام عشوائية ، تختلف عن الأرقام التى حصلنا عليها فى المرة السابقة .

نلاحظ أن جميع الأرقام العشوائية التى حصلنا عليها حتى الآن كانت عبارة عن كسور تتراوح قيمها مابين (0 و 1) ، ويمكن الحصول على رقم عشوائى صحيح (بدون كسر) يقع مابين 1 , 10 ، وذلك باستخدام العبارة التالية :

```
10    LET    A    =    INT (10 * RND + 1)
```

وسيبين البرنامج التالى كيفية الحصول على أرقام عشوائية صحيحة ذات مدى محدد .

---

* لاداعى هنا للخوض فى كيفية عمل هذه التعليمة ؛ إذ أن لها معانى إحصائية تؤثر فى الخطوات الموصلة إلى الرقم العشوائى .

مثال (٩ ــ ٣) :

الهدف : الحصول على ثلاث مجموعات من الأرقام العشوائية الصحيحة التى تقع مابين :

أ ــ 10 , 1

ب ــ 75 , 15

جـ ــ 95 , 80

وكل مجموعة تحتوى على الأرقام العشوائية .

أولاً ــ خطوات الحل :

١ ــ إسناد القيمة الأولية (1) لعداد الأرقام العشوائية .

٢ ــ الحصول على الرقم العشوائى فى المجموعة الأولى .

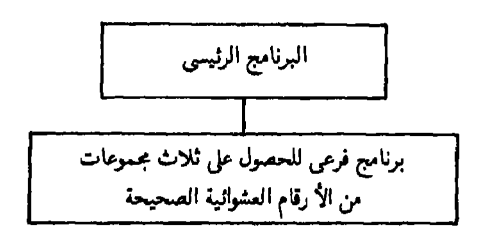
٣ ــ الحصول على الرقم العشوائى فى المجموعة الثانية .

٤ ــ الحصول على الرقم العشوائى فى المجموعة الثالثة .

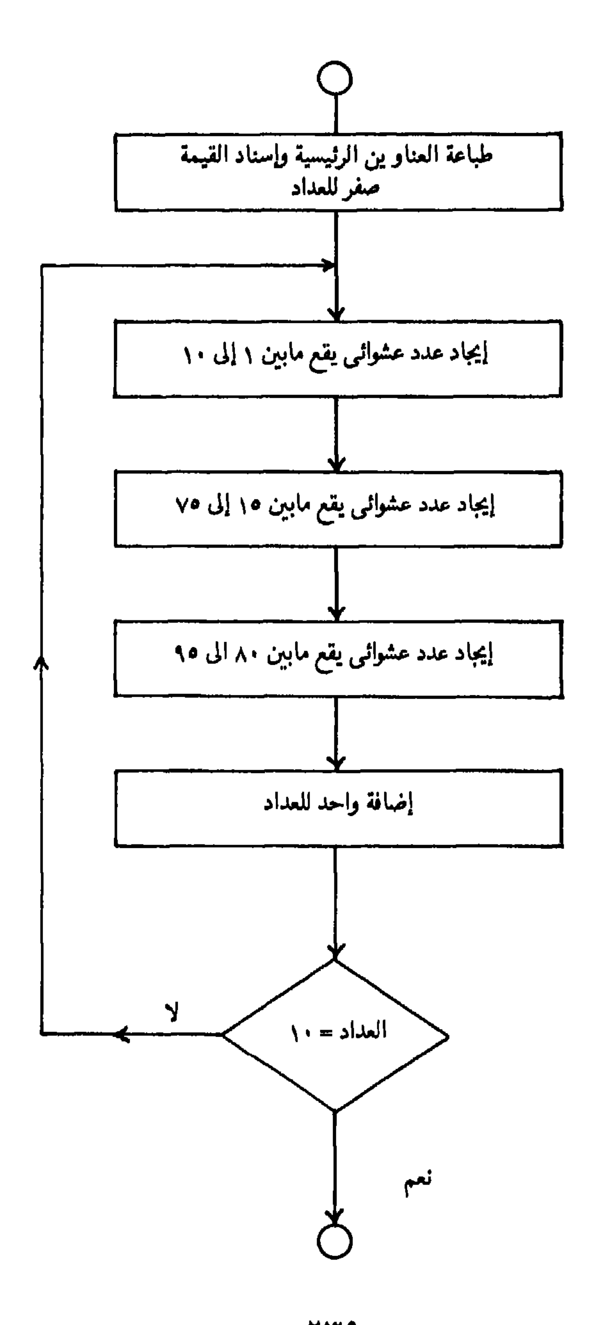
٥ ــ طباعة الأرقام الثلاثة وزيادة العداد بواحد .

٦ ــ الاستفسار عن العداد ، إذا كان أقل من ١١ نكرر الخطوات من (٢ ــ ٥) وإلا نتوقف .

ثانياً ــ الهيكل الهرمى للبرنامج :



ثالثاً ـ نمط البرمجة التركيبية :



رابعاً ــ البرنامج فى الشكل التالى :

شكل (٩ ــ ٣)

برنامج للحصول على ثلاث مجموعات من الأرقام العشوائية الصحيحة

```
10 REM   استدعاء برنامج فرعي
20 GOSUB 40
30 END
40 REM   برنامج فرعي للحصول على ٣ مجموعات من الا رقام
50 REM   طباعة العناوين الرئيسية
60 PRINT TAB(30);" ثلا ثة مجموعات من الا رقام العشوائيه "
70 PRINT TAB(10);"95-80";TAB(20);"75-15";TAB(30);"10-1"
80 PRINT TAB(10);"_____";TAB(20);"_____";TAB(30);"_____"
90 REM   اسناد القيمه الا وليه للعداد
100 LET I = 1
110 LET N =RND
120 REM 10-1   للحصول على قيمه تقع ما بين
130 LET A = INT(N*10)
140 REM 75-15  للحصول على قيمه تقع ما بين
150 LET N =RND
160 IF N < 15/75 THEN 150 : REM
170 LET B = INT(75*N)
180 REM 99-80  للحصول على قيمه تقع ما بين
190 LET N =RND
200 IF N < 80 / 95 THEN 190 : REM
210 LET C = INT(95*N)
220 REM  طباعة الا رقام الثلا ثه التي تم الحصول عليها
230 PRINT
240 PRINT TAB(10);C;TAB(20);B;TAB(30);A
250 LET I = I + 1
260 REM  للا ستفسار عن نهاية عداد الا رقام
270 IF I <= 10 THEN 110
280 RETURN
```

وفى حالة تنفيذ البرنامج ، تظهر النتائج كالتالى :

```
                ثلا ثة مجموعات من الا رقام العشوائيه
95-80       75-15       10-1
-----       -----       -----
 82          48          1
 90          59          7
 92          39          7
 88          71          3
 94          42          5
 89          49          2
 89          66          3
 88          30          3
 88          38          2
 86          41          6
```

مثال (٩ ـ ٤) :

يصل الزبائن لمحطة بنزين بطريقة عشوائية وبمعدل واحد كل خمس دقائق

(٥ × ٦٠ = ٣٠٠ ثانية) .

الهدف : استخدام نظام التظاهر Simulation للحصول على التالى :

١ ـ إيجاد عدد الزبائن الذين سيصلون للمحطة فى خلال ساعة ما .

٢ ـ إيجاد عدد المرات التى سيكون بها موظف المحطة مشغولاً .

٣ ـ إيجاد عدد الزبائن الذين انتظروا قبل أن تتم خدمتهم .

ملاحظة : سيتم الحصول على رقم عشوائى فى كل دقيقة (٦٠ ثانية). وبما أن وصول الزبائن هو بمعدل واحد كل خمس دقائق للمحطة فإنه :

أ ـ عدد الأرقام العشوائية التى سيتم الحصول عليها يساوى $\frac{٣٠٠}{٦٠}$ = ٥

ب ـ يمكن استخدام القيمة من (صفر إلى $\frac{٦٠}{٣٠٠}$ أو ٢,) كمؤشر لوصول زبون للمحطة .

أولاً ـ خطوات الحل :

١ ـ الحصول على الرقم العشوائى .

٢ ـ الاستفسار عن وصول زبون للمحطة ، وذلك بمقارنة قيمة الرقم العشوائى بالقيمة ٢,٠ .

٣ ـ الاستفسار عن نهاية الخدمة للزبون .

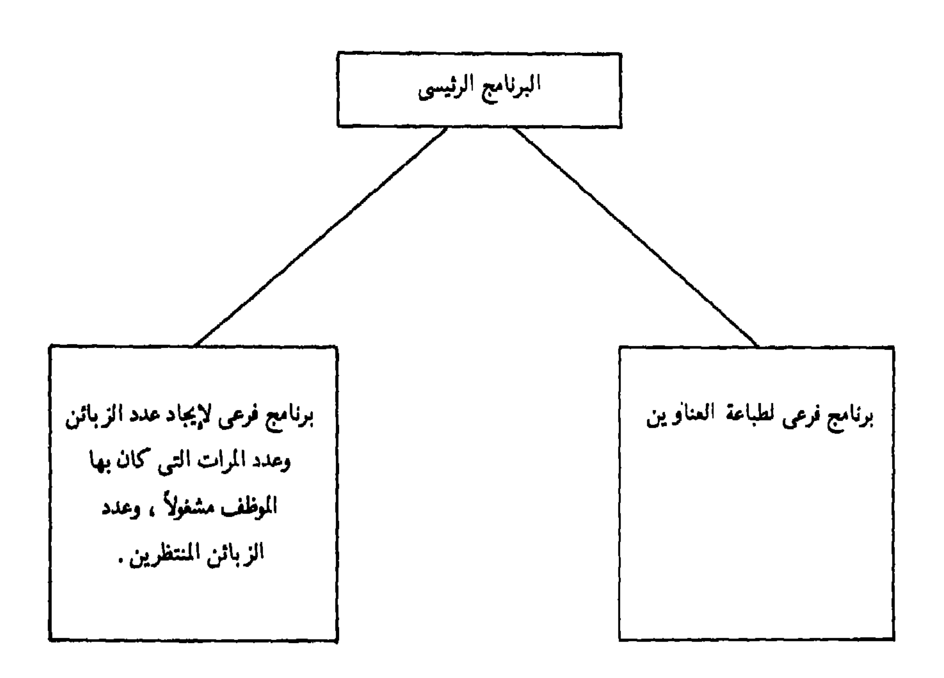
٤ ـ الاستفسار عن عدد الأشخاص المنتظرين للخدمة .

٥ ـ إخراج الوقت ، الاستفسار عن وصول زبون ، الحالة التى بها الموظف وقت نهاية الخدمة ، عدد الأشخاص المنتظرين .

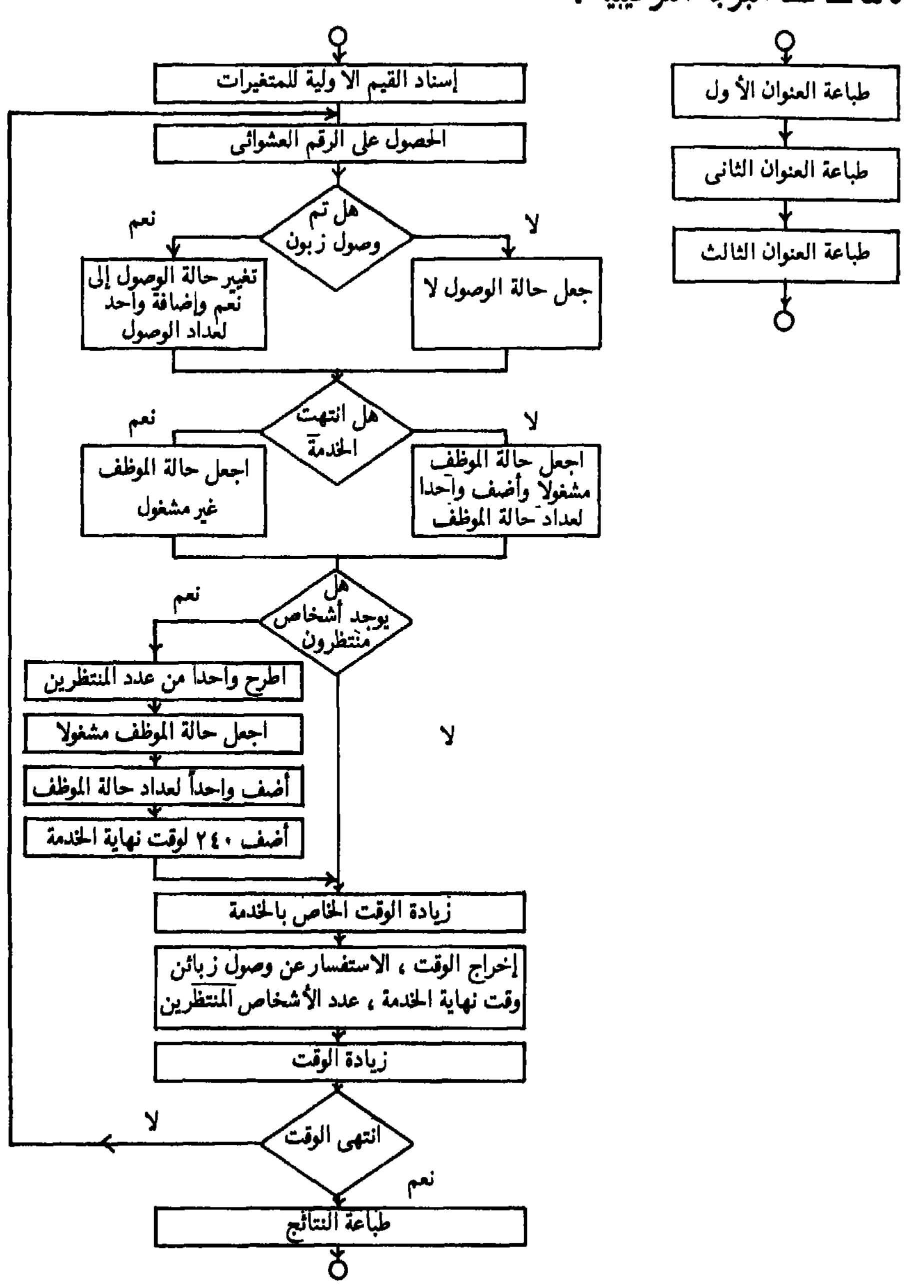
٦ ـ تكرار الخطوات من (١ ـ ٥) إلى أن تنتهى الفترة الزمنية المرادة (٦٠ دقيقة) .

٧ ـ إخراج عدد الزبائن الذين وصلوا للمحطة ، عدد المرات التى كان بها الموظف مشغولاً ، عدد الزبائن الذين انتظروا .

ثانياً ـ الهيكل الهرمى للبرنامج :



ثالثاً ــ نمط البرمجة التركيبية :



رابعاً ــ البرنامج فى الشكل التالى :

شكل (٩ ــ ٤)

برنامج خاص بعمليات انتظار الزبائن أمام محطات البنزين

```
10 REM   استدعاء برنامج فرعي
20 GOSUB 50
30 REM   استدعاء برنامج فرعي
35 GOSUB 100
40 END
50 REM  برنامج فرعي لطباعة العناوين الرئيسية
60 PRINT TAB(20);"محطة بنزين رقم  1  "
70 PRINT"الوقت    هل وصل زبون  حالة الموظف  نهاية الخدمه    عدد المنتظرين "
80 PRINT"  -------------     -----------    ---------     ----------     ----------"
90 RETURN
100 REM  برنامج فرعي للا ستخدام نظام التظاهر
110 REM   اسناد القيم الا وليه للمتغيرات
120 LET W = 0
130 LET S = 0
140 LET A$ = "     "
150 LET E$ = "غير مشغول"
155 LET P$="     ###         ####.##        \          \      \          \     ####"
160 LET T = 0
170 REM   الحصول على الرقم العشوائي
180 LET R = RND
190 REM    الا ستفسار عن وصول زبون
200 IF R > .2 THEN 245
210 REM    اضافه واحد لعداد المنتظرين
220 LET W = W + 1
230 LET A$ = "نعم "
240 REM    الا ستفسار عن عدد المنتظرين
242 GOTO 250
245 REM   تغيير حالة الوصول الى لا
247 LET A$ = " لا "
250 IF T >= S THEN 280
260 REM    لطباعة النتائج و الحاله
270 GOTO 360
280 REM   اسناد فراغ لحالة الموظف
290 LET E$ = "      "
300 REM  الا ستفسار عن عدد المنتظرين
310 IF W < 1 THEN 350
320 REM   طرح واحد من عدد المنتظرين
330 LET W = W - 1
340 LET E$ = "مشغول "
350 LET S = T + 240
360 PRINT TAB(8);W;TAB(19);S;TAB(34);E$;TAB(51)A$;TAB(67);T
370 LET T = T + 60
380 REM   للا ستفسار عن نهاية البحث و التظاهر
390 IF T <= 1200 THEN 180
395 PRINT "================================================================
==="
400 RETURN
```

وفى حالة تنفيذ البرنامج السابق ، سنحصل على النتائج التالية :

محطة بنزين رقم 1

الوقت	هل وصل زبون	حالة الموظف	نهاية الخدمه	عددالمنتظرين
0	نعم	مشغول	240	0
60	لا	مشغول	240	0
120	لا	مشغول	240	0
180	لا	مشغول	240	0
240	لا		480	0
300	نعم		480	1
360	لا		480	1
420	لا		480	1
480	نعم	مشغول	720	1
540	لا	مشغول	720	1
600	لا	مشغول	720	1
660	لا	مشغول	720	1
720	لا	مشغول	960	0
780	لا	مشغول	960	0
840	لا	مشغول	960	0
900	لا	مشغول	960	0
960	لا		1200	0
1020	لا		1200	0
1080	لا		1200	0
1140	لا		1200	0
1200	لا		1440	0

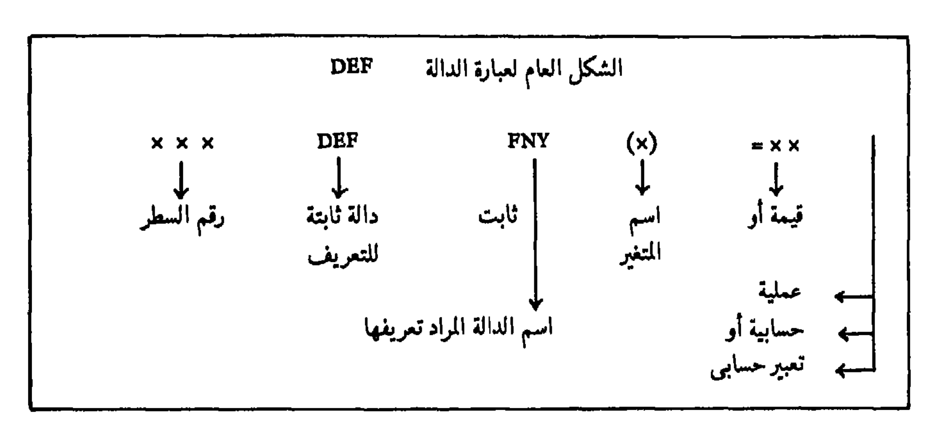
==

الدالات المعرفة من قبل المبرمج

مقدمة عن الدالات المعرفة :

حتى الآن تم التعرض لبعض الدالات الجاهزة Built - in Functions والتى صممت وكتبت بواسطة الشركات المصنعة للأجهزة ، لكن توجد هناك حالات أو عمليات يرغب المستخدم فى القيام بها ، فى أكثر من خطوة فى البرنامج ، أو لأكثر من جزء من البيانات .

يمكننا عمل ذلك بتحديد نوع العمليات المراد إجراؤها باستخدام التعليمة DEF وهى عبارة عن دالة جاهزة. يمكننا تعريف العمليات المرادة وإسنادها إلى اسم الدالة ، ومن ثم استدعاؤها فى أى جزء من أجزاء البرنامج ، كما هو الحال فى تعليمة GOSUB وذلك حسب الشكل التالى :



حيث إن :

— DEF تستخدم كمؤشر للحاسب بأن مايليها هو تعريف لدالة .

— FN اختصار لـ Function Name وتستخدم كمؤشر للحاسب بأن الحرف الذى يليها هو اسم الدالة المراد تعريفها .

Y ــ اسم الدالة، ويجب أن يكون أحد الحروف الأ بجدية اللا تينية من A إلى Z .

(X) ــ اسم المتغير الذى سيحوى القيمة المراد معالجتها عند استدعاء الدالة المعرفة .

X X ــ يمكن أن يكون قيمة أو عملية حسابية (جمعاً ، ضرباً ، قسمة ...) أو تعبيراً حسابياً (معادلة جبرية ....) .

تطوير برنامج باستخدام الدالة المعرفة :

مثال : (٩ ــ ٥)

الهدف : باستخدام تعليمة الدالة DEF ، إيجاد ما يماثل الدرجات المئوية بالفهرنهايتية وبالعكس.

أولاً ــ خطوات الحل :

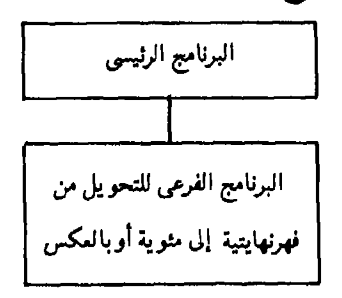
١ ــ تعريف دالة التحويل من المئوية إلى الفهرنهايتية .

٢ ــ تعريف دالة التحويل من الفهرنهايتية إلى المئوية .

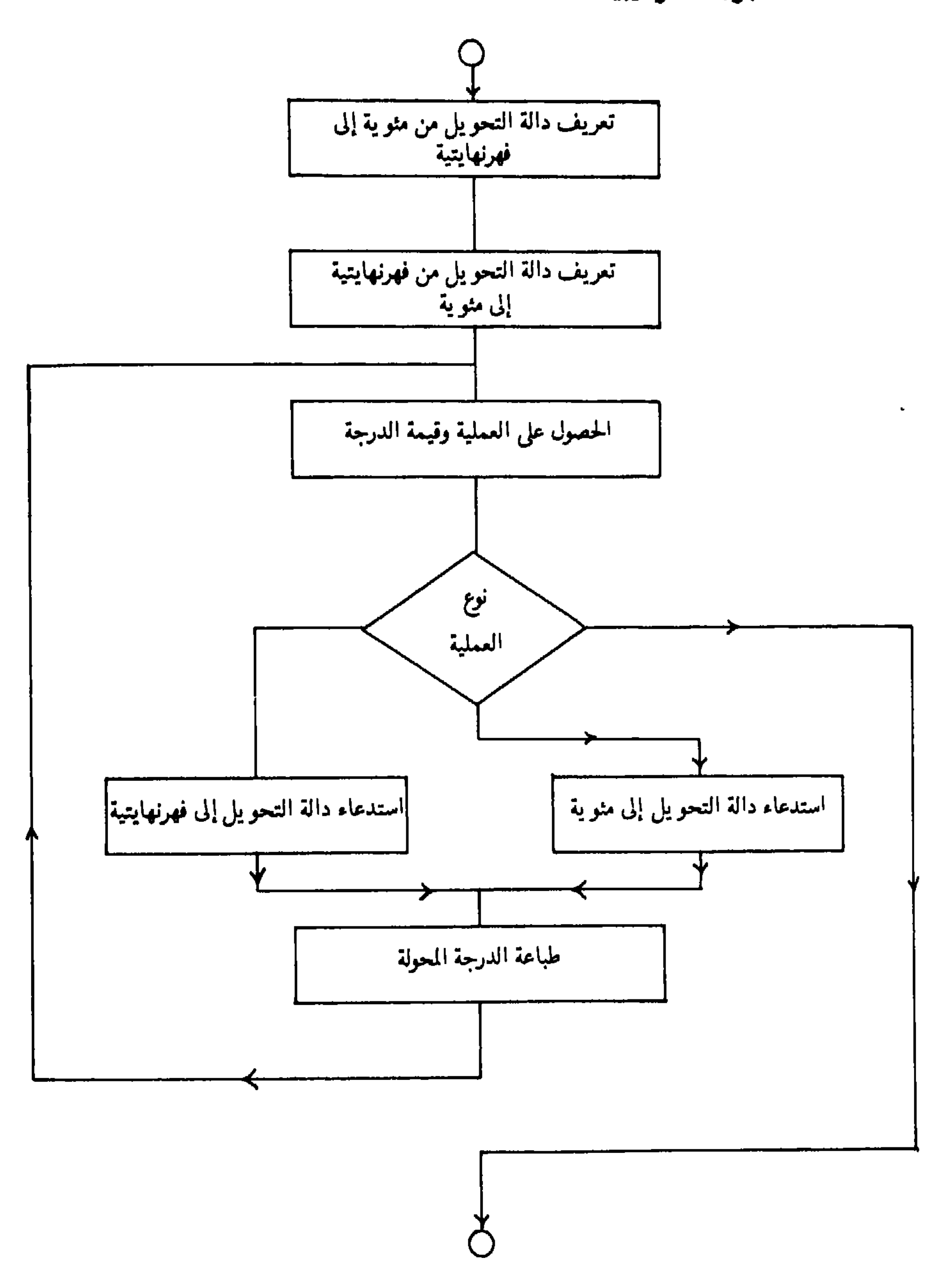
٣ ــ الحصول على نوع العملية وقيمة الدرجة .

٤ ــ استدعاء الدالة حسب نوع العملية ، ومن ثم طباعة مايماثلها .

ثانياً ــ الهيكل الهرمى للبرنامج



ثالثا ـــ نمط البرمجة التركيبية :



رابعاً ــ البرنامج فى الشكل التالى :

شكل (٩ ــ ٥)

برنامج لتحويل الدرجات المئوية إلى فهرنهايتية وبالعكس باستخدام الدالة المعرفة DEF

```
10 REM استدعاء برنامج فرعي
20 GOSUB 40
30 END
40 REM برنامج فرعي لتحويل الدرجات المئويه الى فهرنهايتيه وبالعكس
50 REM تعريف داله التحويل من مئويه الى فهرنهايتيه
60 DEF FNF(T) = (((9/5) * T) + 32)
70 REM تعريف داله التحويل من فهرنهايتيه الى مئويه
80 DEF FNC(T) = (5/9) * (T - 32)
90 REM للحصول على نوع العمليه وقيمه الدرجه
100 PRINT TAB(35);"لائحه الاختيارات الرئيسيه"
105 PRINT
110 PRINT TAB(45);"الرمز      نوع العمليه"
120 PRINT TAB(45);"-----     ------------"
125 PRINT
130 PRINT TAB(30);" 1   للتحويل من فهرنهايتيه الى مئويه"
140 PRINT TAB(30);" 2   للتحويل من مئويه الى فهرنهايتيه"
150 PRINT TAB(30);" 3   انهاء العمليه والتوقف"
160 PRINT TAB(30);"  ----------------------------------"
165 PRINT
170 INPUT "ادخل رمز العمليه المطلوبه والدرجه مفصولتان بفاصله";N,D
180 REM للاستفسار عن نوع العمليه المطلوبه
190 ON N GOTO 250,230,270
200 IF N >= 1 AND N < 3 THEN 100
210 PRINT "--------- خطا في الادخال ---------"
220 GOTO 170
230 LPRINT "فهرنهايتيه";FNF(D) ;" درجه مئويه تساوي";D
240 GOTO 100
250 LPRINT "مئويه";FNC(D);"درجه فهرنهايتيه تساوى ";D
260 GOTO 100
270 RETURN
```

وفى حالة تنفيذ البرنامج السابق سنحصل على النتائج التالية :

أ ــ شاشة الاختيارات :

لائحة الاختيارات الرئيسية

الرمز	نوع العملية
1	للتحويل من فهرنهايتية إلى مئوية .
2	للتحويل من مئوية إلى فهرنهايتية .
3	إنهاء العملية والتوقف .

أدخل رمز العملية المطلوبة والدرجة مفصولتين بفاصلة

ب ــ وعند اختيار «1» وطباعة درجة الحرارة بالفهرنهايتية 60 :

60 درجة فهرنهايتية تساوى 15.55556 مئوية

جـ ــ وعند اختيار «2» وطباعة درجة الحرارة بالمئوية 32 :

32 درجة مئوية تساوى 89.6 فهرنهايتية .

فى المثال شكل (٩ ــ ٥) تم استخدام الدالة DEF واستدعاؤها للتعامل مع قيمة واحدة ، وبنفس الطريقة يمكننا التعامل مع أكثر من قيمة واحدة ، فعند إيجاد مساحة مستطيل ، فالخطوات والعبارات ستكون كالتالى :

١ ــ تعريف الدالة لإيجاد المساحة .
٢ ــ الحصول على قيمتى الطول والعرض .
٣ ــ استدعاء الدالة لحساب المساحة .

```
10    DEF FNM (A , B) = A * B
20    LET C = FNM (5 , 8)
30    PRINT C
40    END
```

تمارين

١ ــ اعمل على طباعة وتنفيذ البرامج الموجودة فى أشكال الفصل وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعديلات.

٢ ــ بيّن الأخطاء إن وجدت فى كل من العبارات التالية :

أ
```
10   PRINT    SIN 20
```
ب
```
10   LET      A $ = COS (40)
```
جـ
```
10   DEF      FNA (B) = B * 2 + 20
```
د
```
10   DEF      (X1 , X2) = (X1 ** 2) - (X1 / X2)
```
هـ
```
10   LET      A = PI + SQR (ABS (20)
```

٣ ــ بين نتائج تنفيذ البرامج التالية :

أ ــ
```
10   LET      I = — 2
20   IF       I > 3 THEN 60
30   PRINT    INT (I) , INT (I + .5)
40   LET      I = I + .5
50   GOTO 20
60   END
```

ب ــ

```
10   DEF     FNA (B) = (B * 1) + 2
20   READ    B
30   IF      B< 0 TEHEN 70
40   PRINT   FNA (B)
50   DATA    10 , 5 , 2 , - 1 , 4
60   GOTO 20
70   END
```

٤ ــ اكتب برنامجاً يعمل على طباعة جيب الزاوية للزوايا من عشر إلى أربعين .

٥ ــ باستخدام تعليمة DEF اكتب برنامجاً يعمل على الحصول على عشرة أرقام عشوائية ، ومن ثم إيجاد تربيع وتكعيب الرقم .

٦ ــ اكتب برنامجاً يعمل على إيجاد عشرة أرقام عشوائية تقع بين مائة ومائة وخمسين .

٧ ــ اكتب برنامجاً باستخدام الدالة DEF يعمل على :

أ) التحويل من بوصة إلى سنتيمتر وبالعكس .

ب) التحويل من أميال إلى كيلومترات وبالعكس .

جـ) التحويل من كجم إلى باوند وبالعكس .

مع العلم بأن :

البوصة = ٢,٥٤ سم ، كجم = ٢,٢٥ باوند، الميل = ١,٦ كم.

٨ ــ اكتب برنامجاً باستخدام الدالة DEF يعمل على الحصول على ثلاث قيم، ومن ثم إجراء العمليات الحسابية من جمع وضرب عليها.

الفصل العاشر

التحكم فى طبع البيانات المخرجة والمعلومات

استعمالات تعليمة اطبع باستخدام PRINT USING :

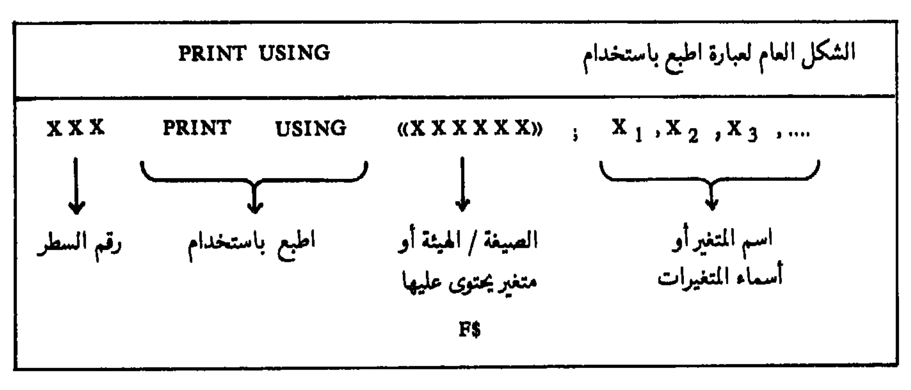
تعرضنا فى الفصول السابقة لكيفية التحكم إلى حد ما فى صيغة النتائج المعالجة ، والمراد إخراجها باستخدام الفاصلة (,) أو الفاصلة المنقوطة (;) أو تعليمة () TAB . وكان هذا التحكم فى العمود الذى سيبدأ بعده طباعة القيمة التى يحتويها المتغير ، كما نلاحظ فى المثال التالى الذى يقرأ و يطبع خمس قيم :

```
10 REM  استدعاء برنامج فرعي
20 GOSUB 40
30 END
40 REM  اظهار كيفية التحكم في الطباعة
50 REM (اعمده) لطباعة العناوين
60 PRINT "=================================================="
70 PRINT "         1         2         3         4         5"
80 PRINT "12345678901234567890123456789012345678901234567890"
90 PRINT "=================================================="
100 REM  اسناد قيم اوليه للمتغير
110 LET I = 0
120 REM  قراءة القيمه العدديه
130 READ A
140 PRINT TAB(5);A
150 REM اضافة واحد للعداد
160 LET I = I + 1
170 REM  استفسار عن قيمة العداد
180 IF I < 5 THEN 130
190 PRINT "=================================================="
200 REM  البيانات
210 DATA 45,107,7,320,1320
220 RETURN
```

وفى حالة تنفيذ البرنامج السابق نجد أن المخرجات ستظهر كالتالى :

```
==================================================
        1         2         3         4         5
12345678901234567890123456789012345678901234567890
==================================================
     45
    107
      7
    320
   1320
==================================================
```

نلاحظ أن جميع القيم بدأت طباعتها فى العمود السادس كما حدد لها فى عبارة رقم 140 ، فلو أردنا إخراج هذه القيم بطباعة كل رقم من كل قيمة فى الخانة الخاصة به (خانة الآحاد ، العشرات ، المئات ، الألوف ....) على حسب قواعد النظام العشرى ، أى تحديد الهيئة أو الصيغة (FORMAT) التى ستظهر بها كل قيمة ، يمكننا عمل ذلك باستخدام تعليمة «اطبع باستخدام PRINT USING» ، كما هو موضح فى الشكل التالى :



لاحظ أن الصيغة إما أن تظهر فى عبارة PRINT USING أو تخزن أولاً فى متغير غير حسابى ، ومن ثم يذكر اسم المتغير فى العبارة ، وفى كلتا الحالتين فإن الصيغة/الهيئة ستحتوى مابين علامتى التنصيص « » على رمز خاص أو أكثر ، و يعتمد انتقاء الرموز على نوع البيانات المراد إخراجها (حسابية أو غير حسابية) ، وستتضمن الأمثلة التالية كيفية استخدام هذه التعليمة وفق هاتين الحالتين .

البيانات العددية :

طباعة الأرقام الصحيحة : يستخدم الرمز الخاص «#» لتخصيص كيفية طباعة الأرقام الحسابية . فلإخراج الأرقام السابقة ، كل فى خانته الخاصة به ، نعمل على تعديل عبارة 140 فى البرنامج السابق فتصبح كالتالى :

```
140    PRINT USING    « # # # # # # # » ; A
```

حيث تم تحديد الحجم الذى ستطبع به كل قيمة بسبع خانات ، كما هو مبين فى تعليمة رقم (140) ، وفى حالة تنفيذ البرنامج ستظهر المخرجات كالتالى :

```
==================================================
          1         2         3         4         5
12345678901234567890123456789012345678901234567890
==================================================
   45.000
  107.000
    7.000
  320.000
 1320.000
==================================================
```

ونجد أن القيمة قد طبعت من اليمين إلى اليسار ، ومن ثم وضع كل رقم من كل قيمة فى الخانة الخاصة به .

أما استخدام الأسلوب الآخر فيتطلب ذلك تعديل السطر (140) وإضافة السطر (105) كالتالى :

```
105    LET    A$   =   « # # # # # # # »

140    PRINT  USING  A$ ; A
```

طباعة قيم عددية تحتوى على كسور عشرية : فى حالة وجود قيم عددية كسرية يمكننا تعديل الصيغة فى نفس العبارة (140) ، حيث تحدد عدد الخانات الكسرية المراد

طبـاعتـهـا إلى يمين الـفـاصـلـة العشرية («.» باللغة الانجليزية) وعدد الخانات للعدد الـصحيح إلى يسارها . ففى المثال السابق ، إذا افترضنا وجود بيانات رقمية تحتوى على كسور عشرية كالتالى :

```
210    DATA    45.0,    107.34.7.9,    320,    1320.714
```

يجب تـعـديل السطر 140 لتحديد عدد خانات العدد الصحيح وعدد خانات الكسر العشرى ، ليصبح كالتالى :

```
140 PRINT USING   « ######·### » ; A
```

وفى حالة تنفيذ البرنامج ، ستظهر المخرجات كالتالى :

```
==================================================
          1         2         3         4         5
12345678901234567890123456789012345678901234567890
==================================================
     45
     107
     7
     320
     1320
==================================================
```

لاحظ أن البيانات فى عبارة رقم (210) احتوت على قيمة صحيحة (320) . لكن احتواء التخصيص على كسور عشرية (« ### · ##### ») يؤدى إلى إظهار جميع القيم على شكل كسور .

ويمكن تـعديل السطر 140 وإضافة السطر 105 مرة أخرى لتتوافق طباعة النتائج مع الأسلوب الآخر .

```
105    LET    A$   =   « ##### - ### »
140    PRINT  USING   A$   ;   A
```

الرموز الخاصة :

أما طباعة الرموز الخاصة بالقيم المخرجة فيتم كالتالى :

مثال ١ ــ إظهار إشارة القيمة (سواء كانت سالبة ــ أو موجبة +) أمامها . و يتم ذلك بوضع إشارة + أمام الرموز # فى الصيغة كما هو مبين فى العبارة التالية :

```
10 PRINT USING «+ # # # . # # » ; 100 . 3 ,98.5 ,–107.3 , 17.4 ,–.75
```

وعند تنفيذ العبارة ستظهر المخرجات كالتالى :

```
+   100.30   +   98.50   –   107.30   +   17.4   –   .75
```

لاحظ أنه تم وضع فراغين فى الصيغة حتى يتم تفريق طباعة القيم بعضها عن بعض.

مثال ٢ ــ طباعة علامة $ أمام القيم المراد إخراجها ، و يتم بوضع علامتى $$ فى بداية الصيغة كالتالى :

```
20    PRINT USING «$$ # # # . # #   » ; 720.5 , 315.75 , 23.45
```

وعند تنفيذ العبارة ستظهر المخرجات كالتالى :

```
$ 720.50      $ 315.75      $ 23.45
```

مثال ٣ ــ وضع علامة * أمام القيم المراد إخراجها ، و يتم بوضع علامتى * * فى بداية الصيغة كالتالى :

```
30    PRINT USING «** # # # . # #   » ; 1720.5 , 23.45
```

وعند تنفيذ العبارة ستظهر المخرجات كالتالى :

```
*    1720.50        ***  23.45
```

لاحظ أنه تمت تعبئة الفراغات السابقة للقيمة بعلامات * فى حالة كون حجم القيمة أقل من حجم الصيغة المراد استخدامها ، كما هومبين فى طباعة القيمة 23.45 . أما فى حالة كون الرقم المراد طباعته مكوناً من خمسة أرقام ، فلايتم طباعة «*» . أمام الرقم .

مثال ٤ ــ لتسهيل قراءة قيم كبيرة ، يمكننا فصل كل ثلاث خانات (أو أقل أو أكثر) بعضها من بعض بفاصلة كالتالى :

```
40 PRINT USING « #,###,###.## » ; 14578.5 , 479621
```

وعند تنفيذ العبارة ستظهر المخرجات كالتالى :

```
14 , 578.50          479 , 621.00
```

وفى جميع الأمثلة السابقة ، يمكن اتباع الأسلوب الثانى كما فى المثالين السابقين . ففى المثال الأخير مثلاً نعدل السطر 40 ونضيف السطر 30 كالتالى :

```
30    LET    B$ = « #,###,###.## »
40    PRINT  USING    B$ : 14578.5 , 479621
```

البيانات غير العددية :

يتم التحكم فى طباعة البيانات غير الحسابية باستخدام نفس التعليمة PRINT USING ولكن باختلاف الرمز المستخدم ، حيث تستخدم (الشرطة المائلة SLASH /) لتحديد البداية والنهاية (حجم المتغير أو عدد الخانات المراد حجزها حتى تتم طباعة قيم المتغير بها) ، وتحسب الشرطة من ضمن هذا العدد ، فلو أردنا طباعة ١٠ أحرف من متغير ، نترك ٨ خانات فارغة بين الشرطتين .

```
10    READ
15    IF N$ = «X X X» THEN 40
20    PRINT USING «/            /» ; N$
30    DATA «بسام حداد» , «سامى على» , «X X X»
40    END
```

وعند تنفيذ البرنامج ، تظهر النتائج كالتالى :

بسام حداد

سامى على

ويمكن تعديل البرنامج السابق كالتالى ؛ ليتوافق مع الأسلوب الثانى لاستخدام PRINT USING

```
5     LET K$ = «/            /»
20    PRINT USING K$ ; N$
```

وإذا زاد عدد الحروف المخزنة على عدد الحروف المحددة فى عبارة «اطبع باستخدام» يتم تجاهل الزائد المخزن ، أما إذا كان الوضع على العكس من ذلك فيتم ملء الفائض فى عدد الخانات المحدد بالفراغات .

العناوين التعريفية :

يمكن وضع عناوين تعريفية فى تعليمة PRINT USING لإظهارها فى كل مرة يتم فيها استخدام التعليمة ، كما هو مبين فى البرنامج التالى :

```
10 REM  استدعاء برنامج فرعي
20 GOSUB 40
30 END
40 REM اظهار كيفية التحكم في الطباعة  باستخدام تعليمة اطبع باستخدام
50 REM  اسناد قيم اوليه للمتغير
60 LET I = 0
70 REM  f$ اسناد صيغة الطباعة للمتغير
80 LET F$ = "   a =  #####              b =  #####"
90 READ A,B
100 LET I = I + 1
110 PRINT USING F$;A,B
120 DATA 45,9087,4,765
130 REM  استفسار عن عدد مرات الطباعه المراده
140 IF I < 2 THEN 90
150 RETURN
```

وفى حالة تنفيذ البرنامج ستظهر المخرجات كالتالى :

```
a =      45              b =    9087
a =       4              b =     765
```

لاحظ أن « A # » و « B # » قد عوملتا كعنوانين تعريفيين ، ولم تؤثرا على سير البرنامج ، كذلك أخذت بعين الاعتبار، الفراغات الواردة فى الصيغة ، وأخيراً فإن الأسلوب الثانى قد استخدم فى هذا البرنامج .

ملاحظات على استخدام تعليمة PRINT USING :

١ ــ طباعة القيم العددية تتم من اليمين إلى اليسار.

٢ ــ يجب أن يكون عدد أرقام القيم العددية مساوياً أو أقل من عدد الرموز المستخدمة ، وفى حالة زيادة عدد الخانات المراد طباعتها على عدد الخانات المحددة فى التعليمة PRINT USING فتظهر إشارة ٪ على يسار الرقم موضحة هذه الحالة .

٣ ــ يجب أن يكون عدد حروف القيم غير العددية مساوياً أو أقل من عدد المسافات الموجودة بين علامتى \ \.

٤ ــ يستخدم الرمز # للقيم العددية فقط .

٥ ــ يستخدم الرمز \ \ للقيم غير العددية فقط .

٦ ــ عند استخدام علامة * أو علامة الدولار $ يتم حجز خانة الإشارة وتخصم من حجم الصيغة ، لذا يجب أن يكون عدد أرقام القيمة أقل من عدد الرموز بواحد .

٧ ــ يفضل استخدام الأسلوب الأول فى تحديد الصيغة ــ أى إدراجها فى عبارة PRINT USING إذا كانت العبارة ستنفذ مرة واحدة أو أن الصيغة مبسطة ، أما فى حالة استخدام نفس الصيغة فى أكثر من عبارة PRINT USING أو أنها معقدة ، فيفضل استخدام الأسلوب الثانى ــ أى تخزين الصيغة فى متغير حرفى .

٨ ــ لاحظ أن استعمال تعليمة «اطبع باستخدام» يكون أفضل ما يكون عند تحديد الصيغة لطباعة كل سطر على حدة ، وسنبين أسلوباً فيما بعد عن كيفية توجيه الطباعة على الشاشة فى أى مكان منها .

تطوير برنامج باستخدام PRINT USING :

مثال (١٠ ــ ١)

الهدف : تعديل مثال (٨ ــ ١) ليعمل تعليمة اطبع باستخدام PRINT USING بدلاً من TAB .

خطوات الحل : الخطوة الأولى والثانية والثالثة كما فى المثال (٨ ــ ١)

الخطوة الرابعة : البرنامج فى الشكل التالى :

شكل (١٠ ــ ١)

برنامج لحساب الراتب الأساسى وخارج الدوام باستخدام PRINT USING

```
10 REM  للاستدعاء برنامج فرعي
20 GOSUB 70
30 END
40 REM  برنامج فرعي لحساب الراتب الاساسي خارج الدوام و الاجمالي
50 REM  للاسناد القيم البداتيه للمتغيرات و طباعة العناوين
60 LET T = 0: LET T1=0
70 PRINT TAB(2);"EMPLOYEE  NAME";TAB(22);"NO. OF HOURS";TAB(37);"HOUR RATE";TAB(
48);"BASE SALARY";TAB(61);"OVER TIME";TAB(72);"NET PAY"
80 PRINT TAB(2);"==============";TAB(22);"============";TAB(37);"=========";TAB(
48);"===========";TAB(61);"=========";TAB(72);"========"
90 REM  للقراءة اسم الموظف عدد ساعات العمل واجرة الساعه
100 READ N$,H,P
110 REM  للاستفسار من نهاية البيانات
120 IF N$="dummy" THEN 290
130 REM  للاستفسار اذا عمل الموظف خارج دوام
140 IF H =<  40 THEN  O = 0 : N = H * P
                ELSE  O = (H - 40) * (P * 1.5) : N = O + (P * 40)
150 REM  للايجاد الراتب الاساسي
160 LET B = N - O
170 LET T = T + N
180 LET T1 = T1 + O
190 REM  لطباعة البيانات
200 PRINT USING "  \                  \       ##.#         ##.#          ###.#      ##
#.#    #,###.#";N$,H,P,B,O,N
210 REM  للقراءة سجل آخر
220 READ N$,H,P
230 GOTO 120
240 DATA ALI SALIM,42,15
250 DATA AHMED AMER,36,19
260 DATA ali ahmed,60,23
270 DATA tameem ahmed,52,18
280 DATA dummy,0,0
290 PRINT TAB(2);"===================================================================
===============" 
300 REM  لطباعة اجمالي خارج الدوام و اجمالي رواتب الموظفين
310 LET T$ = "                                                    T O T A L S   #,###.#
    #,###.#"
320 PRINT USING T$;T1,T
330 PRINT TAB(2);"===================================================================
==============="
340 RETURN
```

وفى حالة تنفيذ البرنامج السابق، سنحصل على النتائج التالية :

EMPLOYEE NAME	NO. OF HOURS	HOUR RATE	BASE SALARY	OVER TIME	NET PAY
ALI SALIM	42.0	15.0	600.0	45.0	645.0
AHMED AMER	36.0	19.0	684.0	0.0	684.0
ali ahmed	60.0	23.0	920.0	690.0	1,610.0
tameem ahmed	52.0	18.0	720.0	324.0	1,044.0
			T O T A L S	1,059.0	3,983.0

ملخص عن الرموز المستخدمة فى تعليمة PRINT USING :

الرمز والهيئة	شرح لاستخدامه
# # # # #	لطباعة قيم عددية صحيحة قيمتها لا تزيد على 99999 .
# # #.# #	لطباعة قيم عددية غير صحيحة (كسور) قيمتها لا تزيد على 999.99 .
###,###.##	لطباعة قيم عددية غير صحيحة ، مع وضع فاصلة بين الأرقام الثلاثة التى توجد على يسار الفاصلة العشرية (.) ، وما بعدها من أرقام وقيمتها لا تزيد على 999,999.99 .
$$ #,###.##	لطباعة قيم عددية غير صحيحة مسبوقة بعلامة الدولار $ مع وجود فاصلة بين الأرقام الثلاثة ، التى توجد على يسار الفاصلة العشرية (.) وما بعدها من أرقام وقيمتها لا تزيد على 99,999.99 $.
# # # # -	لطباعة قيم عددية صحيحة متبوعة بإشارة السالب على يمين القيمة وقيمتها لا تقل عن 9999 .
* * # # #	لطباعة قيم عددية صحيحة مسبوقة بإشارة * وقيمتها لا تزيد على 9999 .
+ # # #	لطباعة قيم عددية صحيحة مسبوقة بإشارة القيمة ، سواء كانت موجبة أو سالبة ، وقيمتها لا تزيد على 99999 .
!	لطباعة الحرف الأول من القيمة غير العددية .
\ ...... \ (1 2 8)	لطباعة قيمة غير عددية وعدد حروفها لا يزيد على ثمانية أحرف . وللاستخدام الأمثل لتعليمات تحديد شكل وصيغ المعلومات المخرجة ، من المفيد أن نتحدث عن تصميم المخرجات والمدخلات .

تصميم المخرجات و المدخلات :

المخرجات :

فى معرض الحديث عن تعليمات التحكم فى طباعة البيانات مثل TAB,PRINT USING,PRINT تم توضيح كيفية طباعة العناوين المعرفة للبيانات ، دون تحديد مسبق لأماكن طباعتها سواء كانت الطباعة على الشاشة أم على الورق ، وعند تصميم برنامج لتطبيق ذى مستوى محدد من التعقيد سيكون من المحتم تصميم المخرجات من البرنامج بشكل مسبق . و يوضح هذا التصميم العمود والسطر فى الصفحة ، أو الشاشة الواحدة والمحدد لبداية ونهاية طباعة كل حقل من البيانات أو من العناوين . وهناك نماذج خاصة لتصميم المخرجات على الورق وأخرى للشاشات . ويحتوى النموذج الأول على ٦٠ سطراً و ١٣٢ عموداً ، ومن الممكن طبعاً استخدام عدد أقل من الأسطر أو الأعمدة أو من كليهما . فى حين يحتوى نموذج تصميم الشاشات على ٨٠ عموداً و ٢٤ سطراً .

وعند استخدام هذه النماذج فى طباعة المعلومات باللغة العربية ، قد تواجه مشكلة بسيطة ، وسبب تلك المشكلة هو التفاوت بين الحروف فى عدد الخانات اللازمة لطباعتها ، وفى كل الأحوال لابأس من الاسترشاد بهذه النماذج ، حيث تزداد الاستفادة منها مع ازدياد الخبرة فى طباعة الحروف العربية . هذا مع العلم بوجود جداول توضح عدد الخانات التى يتطلبها كل حرف .

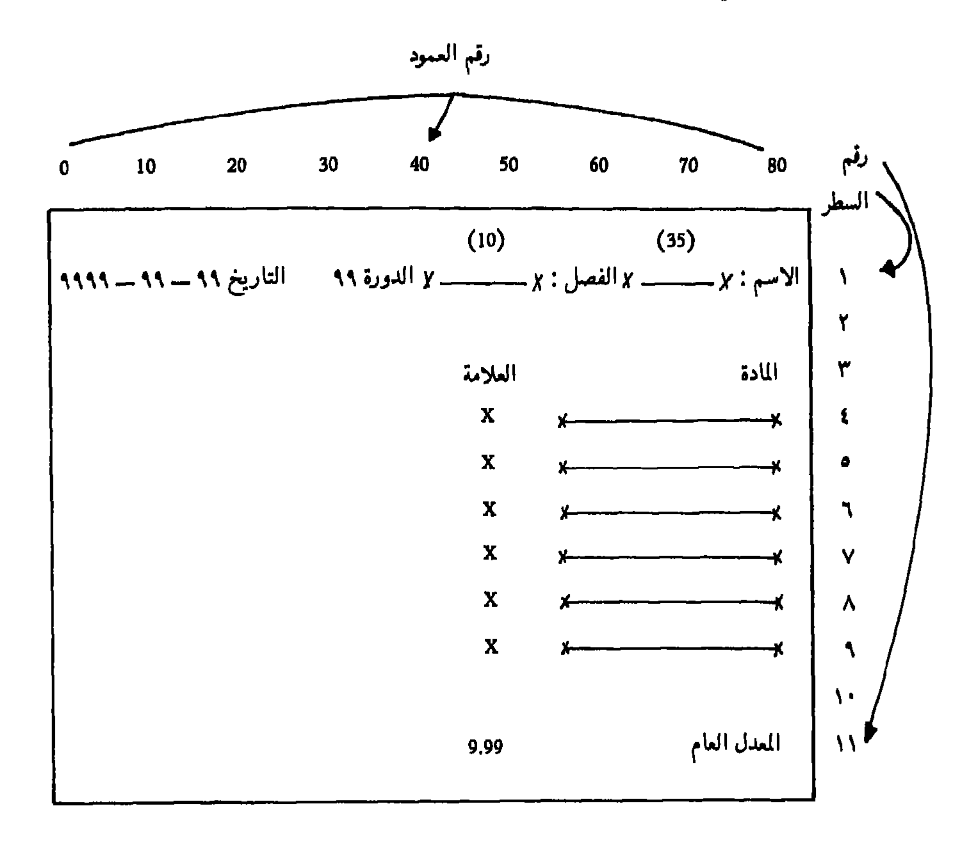
ومن المفيد فى معظم الأحيان تحديد مواصفات البيانات التى ستظهر كمخرجات ، والمثال التالى يبين المواصفات التى ستظهر فى كشف علامات الدارس ، وسيتم استخدام الحرف x للدلالة على البيانات الحرفية والرقم ٩ للدلالة على الأرقام .

مواصفات البيانات فى كشف علامات الدارس

البيانات	الطول	النوع		اسم المتغير
الاسم :	٣٥	حرفى	X (35)	N
الفصل :	١٠	حرفى	X (10)	S
التاريخ :	٨	رقمى	9999-99-99	
	٢	اليوم	99	D
	٢	الشهر	99	M
	٤	السنة	9999	Y
الدورة :	٢	رقمى	99	D1
المواد وعلاماتها متكررة ٥ مرات على الشكل التالى :				
المواد :	٢٥	حرفى	X (26)	S
العلامات :	١	حرفى	X	G
المعدل :	٤	رقمى	9.99	A

(لاحظ حساب خانة واحدة للفاصلة العشرية)

ويحتوى الشكل التالى تصميماً لكشف علامات الدارس :



لاحظ عدم ضرورة تكرار الحرف x ، حيث يذكر العدد فوق الخط الموصل بين بداية ونهاية الحقل .

أما العبارات الخاصة بطباعة هذا النموذج باستخدام تعليمة PRINT USING فهى كالتالى :

العبارة	رقم السطر
10 PRINT USING "#### ##-##-## التاريخ : ## الدورة : / / " ;Y,M,D,D1,S$,N$ الاسم : \ \ الفصل :	1
20 PRINT	2
30 PRINT	3
40 PRINT USING " \\ \ المادة العلامة \ "; S$, G$	4-9
50 PRINT	10
60 PRINT USING " #.## المعدل العام " ;A	11

ويمثل النموذج المرفق الثانى تصميم كشف علامات الدارس على الورق الخاص المعد لهذا الغرض ، أما النموذج الأول فيمثل تصميم نفس الكشف على الورق الخاص بالعرض على الشاشة* .

* عند كتابة برامج شخصية أو عند حل التمارين الموجودة فى هذا الكتاب ، يمكن استعمال الورق المسطر العادى .

I. P. A

VDU SCREEN LAYOUT

PROGRAM : .... DATE ....

MAPSET : ....

MAP NAME : ....

COMMENTS : تصميم كشف درجات على نموذج الشاشة (حجم مصغر)

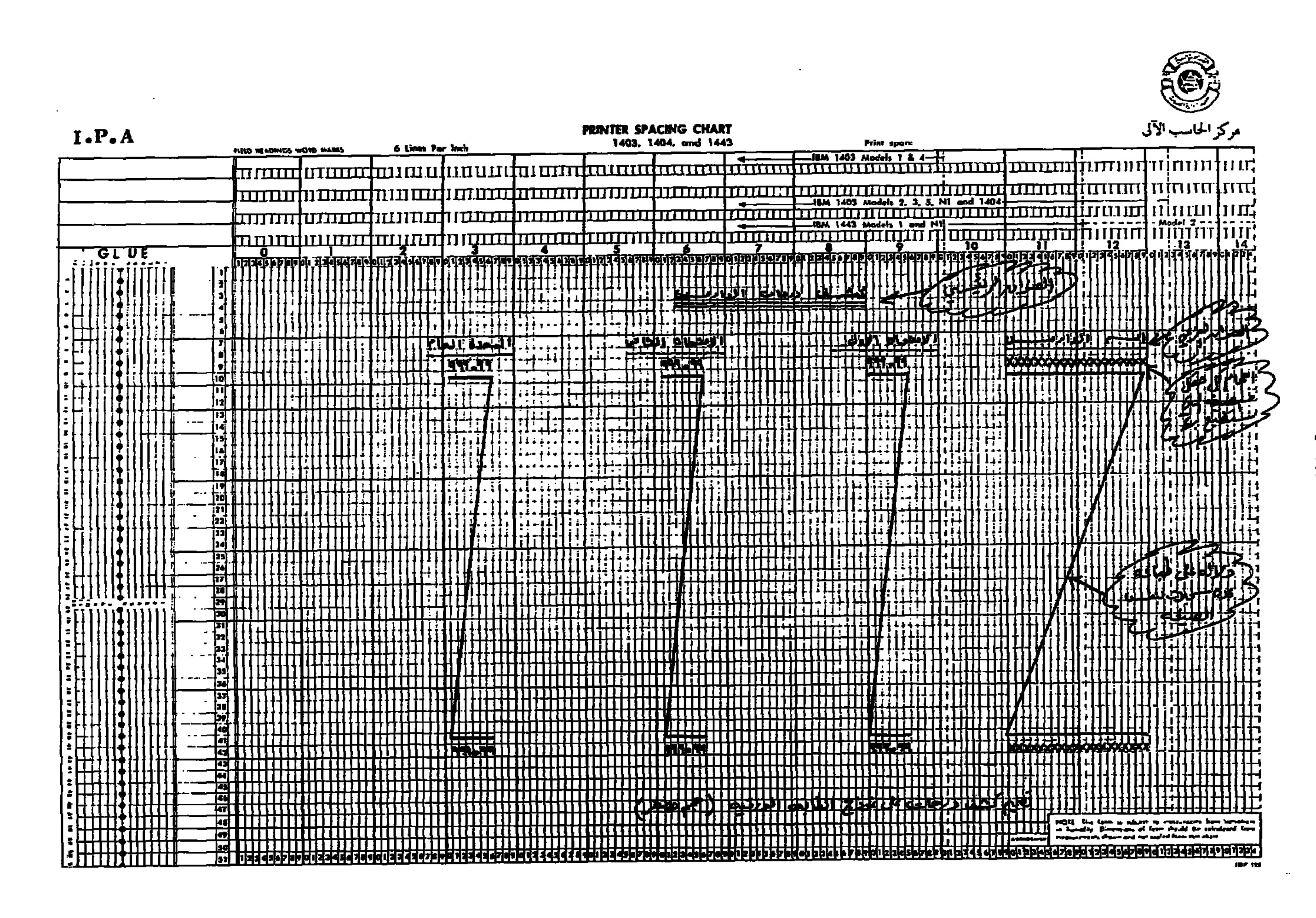
Line	10	30	40	60
3		كشف درجات الدارسين		
4		==========		
6	المعدل العام	الامتحان الثاني	الامتحان الاول	الاسم الدارس
7	------	------	------	------
8	999.99	999.99	999.99	XXXXXXXXXXXXXXXXXXXX
24	999.99	999.99	999.99	XXXXXXXXXXXXXXXXXXXX

العنوان الرئيسي

العناوين الفرعية

احجام كل حقل وصيغته المستخدمة

دلالة على تكرار عدد سجلات بنفس الصيغة



I.P.A
PRINTER SPACING CHART
1403, 1404, and 1443
مركز الحاسب الآلي
6 Lines Per Inch
IBM 1403 Models 1 & 4
IBM 1403 Models 2, 3, 5, N1 and 1404
IBM 1443 Models 1 and N1
GLUE

المدخلات :

تم استخدام تعليمة أدخل INPUT لتسلم البيانات من الشاشة . ولاحظنا أنه عند تنفيذ هذه التعليمة ، يتم إظهار علامة الاستفهام ؟ يعلمنا أن البرنامج يكون فى حالة انتظار للمدخلات ، و يتم إيضاح كيفية طباعة التعليمات الخاصة بالبيانات المدخلة حتى يعلم مستخدم النظام طبيعة هذه البيانات ، و يتم فعل ذلك كله فى المكان الذى يلى تنفيذ آخر تعليمة ، فلو توالى تنفيذ تعليمات INPUT لكان ظهور علامات الاستفهام «؟» متتالية على الشاشة حسب تسلسل خطوات البرنامج ، وقد تتخللها طباعة نتائج جزئية مرحلية .

ويمكن استخدام تعليمة CLS قبل تنفيذ تعليمة INPUT بحيث تظهر علامة الاستفهام ــ سيقوم بأية رسائل إيضاحية أو أوامر للإدخال ــ على رأس الشاشة فى الزاوية اليمنى ...وهكذا ، وكذلك استخدام نفس التعليمة (CLS) قبل تعليمة PRINT يؤدى إلى طباعة المخرجات بدءاً من أول سطر للشاشة ...وهكذا .

وفى هذا الجزء يهمنا نوعان رئيسيان من المؤشرات :

الأول ــ المؤشرات : تأخذ المؤشرات فى العادة قيمة واحدة 4,3,2,1 ... أو حرفاً واحداً A,b,c....X . وهى تحدد الجزء من البرنامج الذى سيتشعب إليه لتنفيذ معالجة معينة . وهذه عادة ما تظهر فى الجزء المتحكم فى عدة أجزاء تنفيذية . و يتم فى العادة تصميم هذه الشاشات ــ شاشات الاختبارات ــ بشكل هرمى يتمشى مع الهيكل الهرمى الوظيفى للبرنامج ، كما تم شرحه فى الفصل الرابع .

الثانى ــ البيانات : حيث يتم تصميم شاشات خاصة باستقبال البيانات . وهذه البيانات المدخلة قد تكون خاصة بالموظف أو بقطع المخزون ، أو بالدارسين ...إلخ .

أما المخرجات من البرنامج فيصمم لها شاشات كما مر ذكره ، وكذلك لطباعة العناوين والبيانات .

التحكم فى استقبال أو طبع المعلومات فى أى مكان على الشاشة :

فالتحكم المطلوب إذن هو إمكانية استقبال أو طبع البيانات فى أى مكان فى شاشة العرض ، أو عند الضرورة ، الطبع فقط على الطابعة وتقسم الشاشة إلى ٢٥ سطراً * و ٨٠ عموداً ، أى أن هناك ٢٠٠٠ موقع يمكن بدء إدخال أو طباعة المعلومات منها . وحتى نفهم كيفية توجيه المؤشر CURSOR إلى أى من هذه المواقع (الخانات) ، يمكن تخيل قلم يتم نقله من موقع لآخر دون أن يؤثر ذلك على ما كتب على الشاشة ، إلا أن تطمس الكتابة السابقة بكتابة جديدة .

والتعليمة التى تستخدم لنقل المؤشر (القلم الوهمى) إلى الخانة المطلوب إدخال البيانات أو طباعتها منها ، هى تعليمة حدد (موقع) LOCATE .

وتتبع الشكل العام التالى :

الشكل العام لعبارة LOCATE			
XX	XX	LOCATE	XXX
↓	↓	↓	↓
٢ ـــ رقم العمود (COLUMN)	(١) رقم السطر (ROW)	التعليمة	رقم السطر
الخلية لإظهار البيانات فيه أو استقبالها منه			

* لاستخدام السطر الخامس والعشرين لابد من إزالة التعليمات الخاصة باستعمال لغة بيسك منها وذلك باستخدام التعليمة التالية قبل محاولة طباعة أى معلومات : XXX KEY OFF

١) يجب ألا يزيد رقم السطر فى حالة استخدام الشاشة على ٢٥ أو ٧٢ فى حالة استخدام الطابعة .

٢) يجب ألا يزيد رقم العمود فى حالة استخدام الشاشة على ٨٠ أو ١٣٢ فى حالة استخدام الطابعة .

ولتوضيح كيفية عمل تعليمة حدد LOCATE ، سيتم طباعة قيمة متغير فى كل من الزوايا الأربع للشاشة ومنتصفها .

فالخطوات المطلوبة هى :

١ ـــ إسناد قيمة المتغير .

٢ ـــ تحديد الزاوية اليسرى العليا وطباعة القيمة .

٣ ـــ تحديد الزاوية اليمنى العليا وطباعة القيمة .

٤ ـــ تحديد الزاوية اليسرى السفلى وطباعة القيمة .

٥ ـــ تحديد الزاوية اليمنى السفلى وطباعة القيمة .

٦ ـــ تحديد منتصف الشاشة وطباعة القيمة .

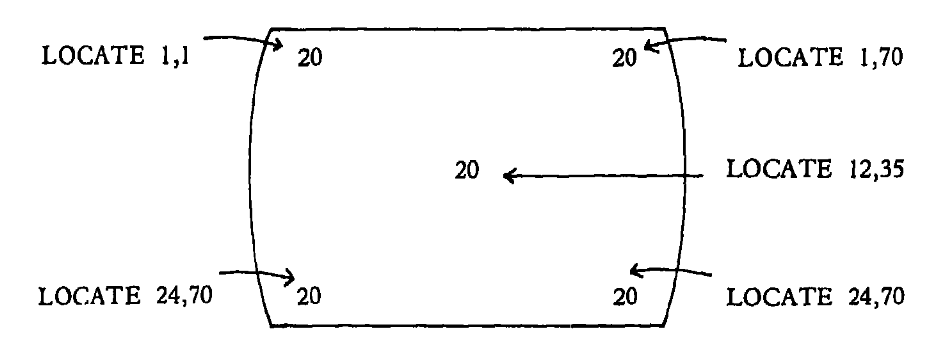
العبارة المستخدمة لكل من الخطوات المذكورة سابقاً :

الخطوة رقم ١	10 LET A = 20
الخطوة رقم ٢	20 LOCATE 1,1: PRINT
الخطوة رقم ٣	30 LOCATE 1,70: PRINT A
الخطوة رقم ٤	40 LOCATE 24,1: PRINT A
الخطوة رقم ٥	50 LOCATE 24,70: PRINT A
الخطوة رقم ٦	60 LOCATE 12,35: PRINT A

وفى حالة تنفيذ هذه التعليمات سيتم إظهار قيمة المتغير A (20) فى الزوايا الأ ربع ، ومـنـتصف الشاشة مع أى محتو يات أخرى كانت ظاهرة على الشاشة سابقاً ، وللحيلولة دون الاحـتـفـاظ بـأية بـيـانات أخرى على الشاشة مع البيانات المراد إظهارها نستخدم تـعليمة CLS (لمسح الشاشة CLEAR THE SCREEN) كما هو مبين فى عبارة رقم 60 فى البرنامج التالى والموضح للخطوات السابقة .

```
10 REM  للا ستدعاء برنامج فرعي
20 GOSUB 40
30 END
40 REM   برنامج فرعي لطباعة قيمه في زوايا الشاشه
50 REM  و من ثم في منتصفها
60 REM  مسح محتويات الشاشه و اسناد قيمه للمتغير
70 CLS: LET A = 20
80 REM  تحديد الزاويه العلويه اليسرى وطباعة القيمه
90 LOCATE 1,1 : PRINT A
100 REM  تحديد الزاويه العلويه اليمنى وطباعة القيمه
110 LOCATE 1,70: PRINT A
120 REM  تحديد الزاويه السفلى اليسرى وطباعة القيمه
130 LOCATE 24,1 : PRINT A
140 REM  تحديد الزاويه السفلى اليمنى وطباعة القيمه
150 LOCATE 24,70: PRINT A
160 REM  تحديد منتصف الشاشه و طباعة القيمه
170 LOCATE 12,35: PRINT A
180 RETURN
```

وفى حالة تنفيذ البرنامج نجد أن البيانات ستظهر كالتالى :



ملاحظة :

يجب أن يؤخذ بعين الاعتبار حجم البيانات المراد إظهارها ، وعلى ضوء ذلك يحدد المكان/المركز ، الذى ستبدأ منه طباعة هذه البيانات ، و يتحقق ذلك عن طريق تصحيح الشاشات كما مر ذكره سابقاً فى هذا الفصل . وكما هو مبين فى العبارتين (100 , 140) كان حجم قيمة المتغير A معروفاً لدينا و يساوى (٢) وعدد خانات الشاشة (٧٢) ؛ لذا نبدأ فى الطباعة من خانة/عمود رقم (٧٠) ، وإلا سيتوقف البرنامج بسبب اجتياز القيمة النهائية لعدد الخانات الممكن استخدامها .

و ينطبق هذا الشرط أيضا على عدد الأسطر الممكن استخدامها ، كما هو مبين فى العبارتين (120 و 140) .

لائحة الاختيارات الهرمية وتصميم الشاشات :

تتسم معظم البرامج المطورة حالياً بخاصية التفاعل والتخاطب فيما بين البرنامج والمستفيد المستخدم ، وذلك عن طريق شاشة العرض ولوحة المفاتيح .

وحيث إن تسلسل تنفيذ الأجزاء الخاصة بأى برنامج يخضع للهيكل الهرمى ، الوظيفى ، فإنه من المحتم تطوير البرنامج التخاطبى عن طريق تصميم شاشات ترتبط فيما بينها بشكل هرمى متناسب ومتوافق مع الهيكل الهرمى للأجزاء .

أنواع ووظائف الشاشات الرئيسية :

أولاً ــ **شاشات لائحة الاختيارات :** وظيفة هذه الشاشات أن تحدد الاختيارات المتاحة للمستفيد ، فى أى مرحلة من مراحل تنفيذ أجزاء البرنامج والرموز التى ستمثل كل اختيار . وستكون هذه الشاشات مرتبة هرمياً حسب تسلسل تنفيذ الأجزاء الظاهرة فى الهيكل الهرمى للبرنامج .

و يتم التشعب وفق هذه الخيارات إلى الجزء ذى العلاقة ، والذى من الممكن أن يقوم بواحدة أو أكثر من الوظائف التالية : عرض لائحة اختيارات أخرى فى مستوى أقل فى الهيكل الهرمى للشاشات ، طلب إدخال بيانات ، معالجة حسابية أو منطقية أو كلتيهما ، طباعة تقرير إما على الشاشة أو على الورق .

ثانياً ــ شاشات الإدخال : تتطلب هذه الشاشات إدخال بيانات ذات نوعين :

ــ بيانات تفصيلية عن سجل شيء معين ، مثل : قطع المخزون ، الموظف ، الدارس ، كالاسم والرقم والعنوان والدورة ...إلخ .

ــ بيانات للقيام بالاستفسار عن موظف ، مثل طباعة اسم أو رقم الدارس . ومن ثم عرض المعلومات المخزنة عنه بعد البحث عن سجل الدارس المحدد فى ملف الدارسين .

ثالثاً ــ شاشات الإخراج : تقوم هذه الشاشات بطباعة المعلومات الناتجة عن البرنامج ، أو المعالجة ، أو الاستفسارات ذات العلاقة بالبرنامج ، كعرض المعلومات عن قطعة معينة أو إحصائيات ملخصة .

وسنركز فى هذا الفصل على استعمال تعليمة حدد (موقع) LOCATE لعرض شاشات الاختيارات إذ أن استخدام هذه التعليمة فى شاشات الإدخال والإخراج غير فعال .

تطوير برنامج باستخدام لائحة الاختيارات الهرمية :

مثال (١٠ ــ ٢) :

الهدف : تطوير برنامج (مثال ٨ ــ ٥) باستخدام لائحة الاختيارات ، على أن يقوم بالاستفسارات التالية :

ــ الحصول على الدخل الكلى للشركة .

ـــ الحصول على دخل فرع معين .

ـــ النسبة المئوية لدخل كل من الفروع إلى الدخل الكلى للشركة .

أولاً ـــ خطوات الحل :

١ ـــ إظهار لائحة الاختيارات الرئيسية ، والتى بواسطتها يتم اختيار نوع الاستفسار المطلوب .

٢ ـــ إذا كان الاختيار يساوى (١) نحسب دخل كل سيارة من السيارات المؤجرة ، ونضيفه إلى المجموع الكلى للشركة ، ونستمر إلى نهاية البيانات ، ومن ثم نظهر المجموع الكلى للشركة .

٣ ـــ إذا كان الاختيار يساوى (٢) نبحث عن السيارات المؤجرة فى ذلك الفرع ، ونحسب دخل السيارة ، ونضيفه إلى المجموع الكلى لذلك الفرع ، ونستمر إلى نهاية البيانات ، ومن ثم نظهر المجموع الكلى للفرع .

٤ ـــ إذا كان الاختيار يساوى (٣) نحسب دخل كل سيارة ومن ثم نضيف الناتج إلى كل من الدخل الكلى للفرع الذى استؤجرت منه والدخل الكلى للشركة ، ونستمر إلى نهاية البيانات ، ومن ثم نحسب النسبة المئوية بضرب الدخل الكلى لكل فرع فى مائة ، ثم قسمة الناتج على المجموع الكلى لدخل الشركة .

ملاحظة :

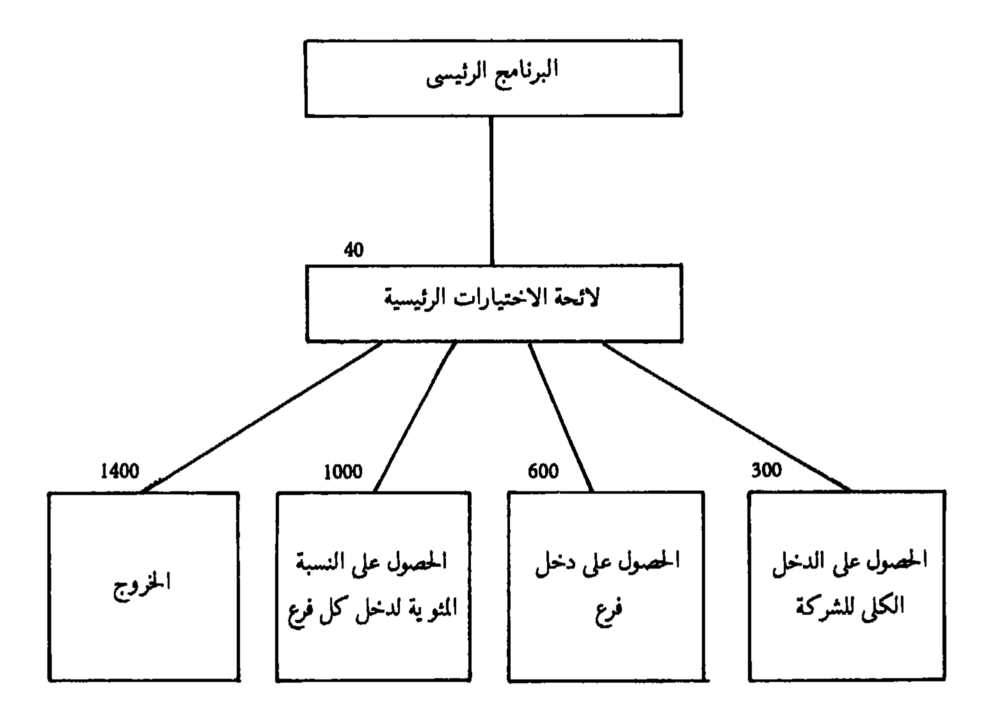
لإجابة كل من الاستفسارات الثلاثة يجب أن نتعامل مع جميع البيانات فى كل مرة يتم فيها الاستفسار عن أى من الحالات .

لكن بعد كل قراءة قيمة من البيانات يتم انتقال المؤشر إلى بداية القيمة التى تليها ، إلى أن يصل إلى نهاية البيانات المزودة فى البرنامج . وإذا أردنا استخدام هذه البيانات فى عملية أخرى يجب أن نعيد المؤشر إلى بداية البيانات المزودة ويمكننا عمل ذلك باستخدام تعليمة (أعد التخزين RESTORE) .

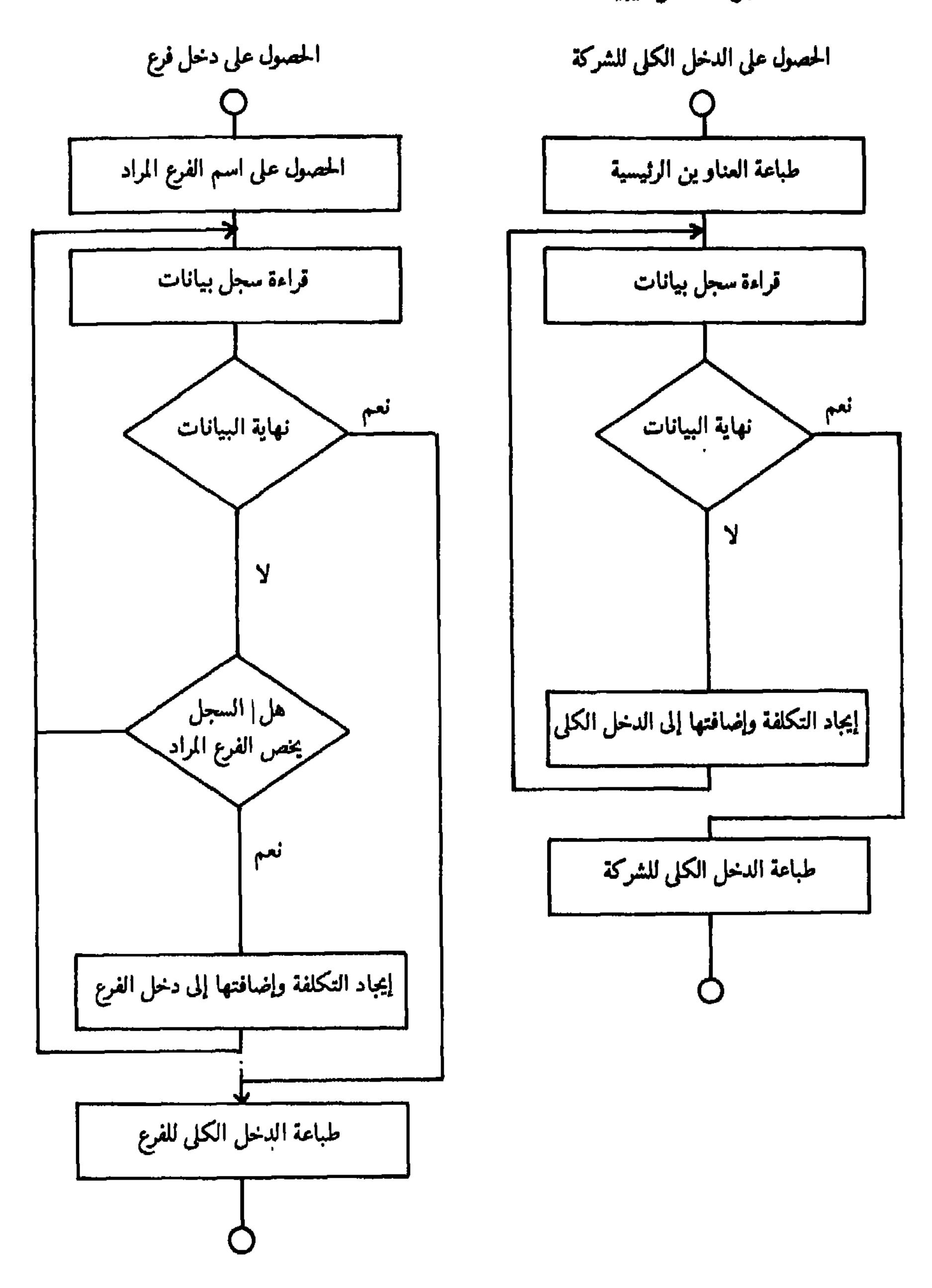
الشكل العام لعبارة	RESTORE
RESTORE	XXX
التعليمة	رقم السطر

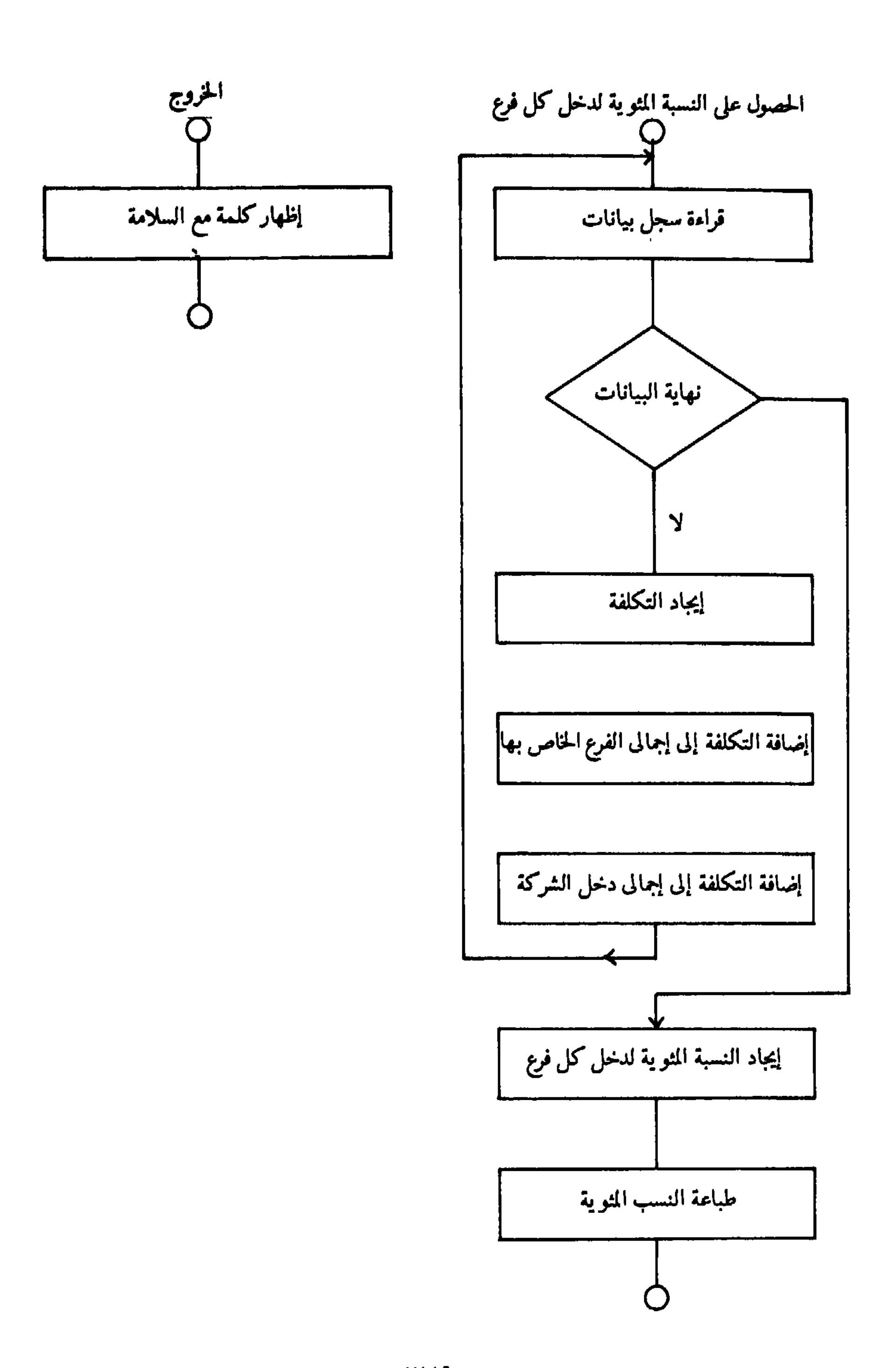
و يتم استخدامها فى البرنامج عندما ننهى قراءة البيانات المراد معالجتها ، ونريد التعامل مع هذه البيانات من بدايتها .

ثانياً ــ الهيكل الهرمى :



ثالثاً ــ نمط البرمجة التركيبية :




الخروج
إظهار كلمة مع السلامة
الحصول على النسبة المئوية لدخل كل فرع
قراءة سجل بيانات
نهاية البيانات
لا
إيجاد التكلفة
إضافة التكلفة إلى إجمالى الفرع الخاص بها
إضافة التكلفة إلى إجمالى دخل الشركة
إيجاد النسبة المئوية لدخل كل فرع
طباعة النسب المئوية

رابعاً ــ البرنامج فى الشكل التالى :

شكل (١٠ ــ ٢)

برنامج لإيجاد الدخل الكلى لكل فرع من فروع الشركة والدخل الإجمالى للشركة

```
10 REM  البرنامج الرئيسي
20 REM  استدعاء برنامج فرعي للاظهار لائحه الاختيارات ومن ثم التشعب
30 GOSUB 50
40 END
50 REM  برنامج فرعي للاظهار لائحه الاختيارات
60 CLS
70  LOCATE 6,30: PRINT "                    لائحه الاختيارات الرئيسيه"
80 LOCATE 7,30: PRINT" -----------------------------------------"
90 LOCATE 8,30: PRINT "            نوع العملية             الرمز"
100 LOCATE 9,30: PRINT" ----------------------------      -----"
110 LOCATE 10,30: PRINT"  للحصول على دخل الشركه الاجمالي       1  "
120 LOCATE 11,30: PRINT"        للحصول على دخل احد الفروع      2  "
130 LOCATE 12,30: PRINT" للحصول على النسبه المئويه لكل فرع     3  "
140 LOCATE 13,30: PRINT"               للانهاء العمليات         4  "
150 LOCATE 14,30: PRINT" -----------------------------------------"
160 LOCATE 16,30: PRINT"         ادخل رمز العمليه المراده"
170 LOCATE 16,30: INPUT N
180 REM  للاستفسار عن نوع العمليه
190 ON N GOSUB 260,660,1050,1460
200 REM  للاستفسار عن نهايه العمليات وصحه البيانات المدخله
210 IF N = 4 THEN 250
220 IF N > 0 AND N < 4 THEN 60
230 LOCATE 20,25: PRINT "*********  خطا في الادخال  ********"
```

```
240 GOTO 170
250 RETURN
260 REM ----------------------------------------------------------------
270 REM  برنامج فرعي للا ظهار لا ئحه الا ختيارات
280 REM ----------------------------------------------------------------
290 REM  لطباعه العناوين
300 CLS:PRINT TAB(30);" الدخل الكلي للشركه"
310 LET T1$=" وجه   اسم  الزبون عدادعند عداد عند كم ايام زياده تكلفه
الل   المدينه"
320 LET T2$="                        الا ستئجار   الا عاده  مستخدمه"
330 LET T3$="======  ====== ===== =======  ========  =========  =====
=== ========"
340 PRINT  " لوجه   اسم  الزبون عداد عند عداد عند كم ايام زياده تكلفه
ال   المدينه"
350 PRINT   "                        الا ستئجار   الا عاده  مستخدمه"
360 PRINT   "======  ====== ===== =======  ========  =========  =====
=== ========"
370 ' PRINT USING T1$: PRINT USING T2$: PRINT USING T3$
380 LET T = 0: REM  اسناد القيمه الا وليه للدخل الكلي
390 RESTORE: REM  وضع المؤشر على بدايه البيانات
400 REM لقراءه البيانات
410 READ C,P,N$,M1,M2,D
420 REM  للا ستفسار عن مؤشر نهايه البيانات
430 IF C = 4 THEN 570
440 GOSUB 1330: REM  استدعاء برنامج فرعي لحساب التكلفه
450 REM  لحساب التكلفه الكليه
460 LET A = A1 + A2
470 REM  اضافه التكلفه للدخل الكلي للشركه
```

```
480 LET T = T + A
490 LET T4$="#####.# #####.#    ##   ####     ######     ###### \   \ ###
##  \      \"
500 REM  للا يجاد اسم المدينه
510 IF C = 1 THEN C$ = "الرياض"
520 IF C = 2 THEN C$ = " جده"
530 IF C = 3 THEN C$ = " الدمام"
540 PRINT USING T4$;A,A2,D,M,M2,M1,N$,P,C$
550 READ C,P,N$,M1,M2,D
560 GOTO 420
570 PRINT "============================================================
=========="
580 DATA 1,13482,"ali    ",1560,3780,3
590 DATA 2,42901,"tameem",6365,7560,5
600 DATA 3,81479,"ahmed ",4712,5104,6
610 DATA 4,00000,"aaaaa ",0000,0000,0
620 PRINT TAB(10); T;" =  مجموع الدخل الكلي للشركه"
630 PRINT "============================================================
=========="
640 FOR W=1 TO 11000:NEXT W
650 RETURN
660 REM -----------------------------------------------------------
670 REM  برنامج فرعي لحساب الدخل الكلي لكل فرع
680 REM -----------------------------------------------------------
690 REM  مسح الشاشه واظهار لا ئحه اختيارات الفروع
700 CLS
```

```
710 LOCATE 9,30:PRINT "=========================================="
720 LOCATE 10,30:PRINT "      لائحة اسماء الفروع المتوفره          "
730 LOCATE 11,30:PRINT "=========================================="
740 LOCATE 12,30:PRINT "      اسم الفرع            الرمز          "
750 LOCATE 13,30:PRINT "  ---------------          -----          "
760 LOCATE 14,30:PRINT "         الرياض              1            "
770 LOCATE 15,30:PRINT "          جده                2            "
780 LOCATE 16,30:PRINT "         الدمام              3            "
790 LOCATE 17,30:PRINT "=========================================="
800 LOCATE 19,30:PRINT "           ادخل رمز الفرع"
810 LOCATE 19,30:INPUT B
820 REM  للتاكد من صحه رمز الفرع المدخل
830 IF B > 0 AND B < 4 THEN 860
840 LOCATE 20,30:PRINT "*********  خطا في الا دخال   *********"
850 GOTO 810
860 REM  وضع المؤشر في بدايه البيانات وقراءه السجل
870 RESTORE:READ C,P,N$,M1,M2,D
880 LET F1 = 0
890 REM  للا ستفسار عن مؤشر نهايه البيانات
900 IF C  = 4 THEN 970
910 IF C <> B THEN 950: REM للا ستفسار اذاكان السجل غير تابع للفرع المطلوب
920 GOSUB 1330: REM  استدعاء برنامج فرعي لحساب التكلفه
930 LET F1 = F1 + A1 + A2
940 REM  قراءه بيانات سجل اخر
950 READ C,P,N$,M1,M2,D
960 GOTO 890
970 REM  طباعه الدخل الكلي للفرع
980 IF B = 1 THEN B$= "الرياض"
```

```
990 IF B = 2 THEN B$= " جده"
1000 IF B = 3 THEN B$=" الدمام"
1010 CLS:LOCATE 12,30
1020 PRINT F1;" = ";B$;" الدخل الكلي لفرع"
1030 FOR W=1 TO 9000:NEXT W
1040 RETURN
1050 REM -------------------------------------------------------
1060 REM  برنامج فرعي لحساب النسبه المئويه لدخل كل من فروع الشركه
1070 REM -------------------------------------------------------
1080 REM  اسناد القيمه الا وليه لكل من مجاميع الفروع الثلا ثه
1090 LET F1=0: LET F2 = 0: LET F3 = 0: LET T = 0
1100 REM  وضع المؤشر في بدايه البيانات وقراءه السجل
1110 RESTORE: READ C,P,N$,M1,M2,D
1120 REM  للا ستفسار عن مؤشر نهايه البيانات
1130 IF N <> 3  AND C = 4 THEN 1320
1140 IF C = 4 THEN 1230
1150 REM  استدعاء برنامج فرعي لحساب التكلفه
1160 GOSUB 1330: LET T = T + A1 + A2
1170 IF C = 1 THEN F1 = F1 + A1 + A2
1180 IF C = 2 THEN F2 = F2 + A1 + A2
1190 IF C = 3 THEN F3 = F3 + A1 + A2
1200 REM  قراءه سجل اخر
1210 READ C,P,N$,M1,M2,D
1220 GOTO 1140
1230 REM  لحساب النسبه المئويه لكل من الفروع الثلا ثه وطباعتها
1240 LET T5$="##.##  = \        \ فرع لدخل النسبه الئويه"
1250 LET P1 = ( F1 * 100 ) / T
1260 LET P2 = ( F2 * 100 ) / T
```

```
1270 LET P3 = ( F3 * 100 ) / T
1280 CLS:LOCATE 10,20:PRINT USING T5$;P1,"الرياض"
1290 LOCATE 11,20: PRINT USING T5$;P2," جده"
1300 LOCATE 12,20: PRINT USING T5$;P3," الدمام"
1310 FOR W = 1 TO 3000: NEXT W
1320 RETURN
1330 REM ----------------------------------------------------
1340 REM           برنامج فرعي لحساب التكلفه لكل سياره مؤجره
1350 REM ----------------------------------------------------
1360 REM  للا يجاد الكيلومترات المستخدمه
1370 LET M = M2 - M1
1380 REM  لحساب التكلفه على حسب عدد الا يام
1390 LET A1 = D * 100
1400 REM لحساب الكيلومترات المسموح استخدامها
1410 LET K = D * 150
1420 REM  لحساب تكلفه الكيلومترات الزائده
1430 LET A2 = 0
1440 IF M > K THEN A2 = ( M - K ) * 3.5
1450 RETURN
1460 REM ----------------------------------------------------
1470 REM  برنامج فرعي لا نهاء العمليات
1480 REM ----------------------------------------------------
1490 REM  مسح الشاشه وطباعه جمله مع السلا مه
1500 CLS
1510 LOCATE 10,30:PRINT "======================"
1520 LOCATE 11,30:PRINT "!      مع السلا مه     !"
1530 LOCATE 12,30:PRINT "======================"
1540 RETURN
```

وفى حالة تنفيذ البرنامج السابق سنحصل على النتائج التالية :

أ ــ إظهار لائحة الاختيارات الرئيسية وإدخال «1» للاختيار الأول :

لائحة الاختيارات الرئيسية

الرمز	نوع العملية
1	للحصول على دخل الشركة الإجمالى
2	للحصول على دخل أحد الفروع
3	للحصول على النسبة المئوية لكل فرع
4	لإنهاء العمليات

أدخل رمز العملية المرادة ? 1

فتظهر الشاشة التالية :

الدخل الكلي للشركة

المدينة	اللوحة	اسم الزبون	عداد عند الاستئجار	عداد عند الإعادة	كم مستخدمة	أيام	زيادة	تكلفة
الرياض	13482	ali	1560	3780	2220	3	6195.0	6495.0
جدة	42901	tameem	6365	7560	1195	5	1557.5	2057.5
الدمام	81479	ahmed	4712	5104	392	6	0.0	600.0

مجموع الدخل الكلي للشركة = 9152.5

ب ــ وإذا دخل رقم «2» من لائحة الاختيارات الرئيسية، تظهر الشاشة التالية :

لائحة أسماء الفروع المتوفرة

الرمز	اسم الفرع
1	الرياض
2	جدة
3	الدمام

أدخل رمز الفرع ? 1

فتظهر الشاشة التالية :

الدخل الكلى لفرع الرياض = 6495

جـ ـ إذا أدخل الرقم «3» من لائحة الاختيارات الرئيسية، تظهر الشاشة التالية :

النسبة المئوية لدخل فرع	الرياض	=	70.96
النسبة المئوية لدخل فرع	جدة	=	22.48
النسبة المئوية لدخل فرع	الدمام	=	6.56

واذا أدخل الاختيار رقم «4» ستظهر الشاشة التالية :

: مع السلامة :

تمارين

١ ــ اعمل على طباعة وتنفيذ البرامج الموجودة فى أشكال الفصل ، وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعديلات .

٢ ــ بين الأخطاء إن وجدت فى كل من العبارات التالية :

أ	`10 PRINT USING "/ /" , A`
ب	`10 PRINT USING "# # , # # #"; Y`
جـ	`10 PRINT USING # # # , B`
د	`10 PRINT USING "# . # # # . # # #"; A,B`
هـ	`10 PRINT USING "— $$ , # # # . # #", R`

٣ ــ لأى من البيانات تستخدم الرموز التالية فى تعليمة PRINT USING

أ #

ب /

جـ *

د !

هـ $

٤ ــ اكتب برنامجاً يعمل على استقبال رقمين من الشاشة ، ومن ثم إظهار لائحة الاختيارات الهرمية لإعطاء المستخدم إمكانية إجراء أى من العمليات الرياضية التالية عليهما :

أ ــ الجمع .

ب ــ الضرب

جـ ـ طرح القيمة الأولى من الثانية.
د ـ طرح القيمة الثانية من الأولى

ومن ثم إظهار ناتج العملية . والاستفسار عن الاستمرارية فى استخدام البرنامج.
ملاحظة : استخدم تعليمة LOCATE

٥ ـ اكتب برنامجاً يعمل على قراءة خمسة سجلات ، مع العلم بأن حقول كل سجل كالتالى : اسم البائع (عشرة أحرف) ، عدد القطع المبيعة من الصنف الأول ، عدد القطع المبيعة من الصنف الثانى ، ومن ثم إيجاد التالى :

أ) الدخل مع العلم بأن :
ـ سعر القطعة من الصنف الأول ٢٠٠ ريال .
ـ سعر القطعة من الصنف الثانى ٢٥٠ ريالاً .

ب) نصيب البائع من الأرباح ، مع العلم بأن النسبة تعتمد على الدخل وموزعة كالتالى :
ـ ١٠٪ إذا كان الدخل أكثر من ١٠٠٠٠ ريال .
ـ ٨ ٪ إذا كان الدخل يتراوح مابين ٨٠٠٠ ـ ١٠٠٠٠ ريال .
ـ ٦ ٪ إذا كان الدخل يتراوح مابين ٥٠٠٠ ـ ٧٩٩٩ ريالاً .
ـ ٤ ٪ إذا كان الدخل يتراوح مابين ٣٠٠٠ ـ ٤٩٩٩ ريالاً .
ـ صفر إذا كان الدخل أقل من ٣٠٠٠ ريال .

جـ) طباعة النتائج كالتالى :
اسم البائع ، عدد القطع المبيعة من الصنف الأول ، الدخل من الصنف الأول ، نصيب البائع من الأرباح ، عدد القطع المبيعة من الصنف الثانى ، الدخل من الصنف الثانى ، نصيب البائع من الأرباح ، إجمالى نصيب البائع من الأرباح .

ملاحظة :

اعمل على اختيار عناوين رئيسية وفرعية مناسبة للتقرير ونتائج المعالجة (أى تصميم نموذج مناسب للطباعة) .

٦ ــ يمكن العودة إلى أى من البرامج فى الفصول السابقة ، وإعادة طباعة المخرجات باستخدام تعليمة PRINT USING .

٧ ــ يحتوى سجل البرامج فى معهد الإدارة العامة على المعلومات التالية :

ــ رقم البرنامج أربعة أرقام

ــ رمز التخصص حاسب آلى 01 بنوك 06

مالية 02 مكتبات 07

ميزانية 03 سكرتارية 08

مستودعات 04

أنظمة 05

ــ مدة التدريب فصل واحد (١) أو فصلان (٢) أو أربعة فصول (٤)

ــ مستوى الدارسين كفاءة 1

ثانوية 2

جامعى 3

ــ الدرجة المعين عليها الدارس ٣ أو ٤ أو ٦

ــ الدرجة المعين عليها بعد التدريب ٤ أو ٨

ــ تاريخ بدء الدورة رقم الفصل ١ ، ٢

السنة ١٤٠x

اليوم xx

الشهر xxx

ـ مجموع الساعات للدورة ٦٠ أو ١٢٠ أو ٤٥

ـ عدد الدارسين فى الدورة ١٠ ـ ٥٠

اعمل على تطوير برامج وفق الهيكل الهرمى للاختيارات للحصول على الإحصائيات التالية :

أ ـ مجموع الدارسين حسب البرنامج (استعمل مصفوفة ذات بعد واحد) .

ب ـ عدد البرامج حسب مستوى الدارسين .

جـ ـ عدد البرامج حسب فصل البداية (الأول أو الثانى) .

استخدم البيانات التالية فى البرنامج ، ويمكن إضافة بيانات أخرى .

1123 , 01 , 1 , 1 , 3 , 4 , 1 , 1407 , 10 , 2 , 60 , 15

1231 , 05 , 3 , 3 , 6 , 8 , 1 , 1407 , 10 , 2 , 60 , 20

1251 . 03 , 2 , 3 , 6 , 7 , 1 , 1407 , 10 , 2 , 45 , 25

1121 , 01 , 4 , 2 , 4 , 6 , 2 , 1408 , 5 , 6 , 120 , 25

2311 , 02 , 2 , 3 , 6 , 7 , 2 , 1408 , 5 , 6 , 45 , 20

3112 , 07 , 2 , 3 , 6 , 7 , 2 , 1408 , 5 , 6 , 45 , 25

1121 , 08 , 2 , 2 , 4 , 6 , 1 , 1407 , 10 , 2 , 60 , 20

الفصل الحادى عشر

المصفوفات ذات البعد الواحد وتعليمات الدوارة البسيطة

مقدمة عن المصفوفات ذات البعد الواحد :

تحدثنا فى الفصول السابقة عن استخدام أسماء المتغيرات لتشير إلى صناديق مستقلة داخل ذاكرة الحاسب ، يتم تخزين البيانات فيها ، ومن الأمثلة على ذلك :

تخزين أسماء الدراسين	N$
تخزين العمر	A
تخزين مساحة المستطيل	S

وفى الأمثلة السابقة لاحظنا أن القيمة الحالية لمحتو يات الصندوق من البيانات تمحى من الذاكرة عند تخزين القيمة الجديدة لها .

ففى المثال البسيط التالى يتم قراءة الاسم وطباعته إلى حين الانتهاء من جميع الأسماء التابعة لدائرة ما .

```
10 READ N$
20 IF N$ = "last" THEN 70
30 PRINT N$
40 DATA "nashid","ahmed","fayez","last"
50 READ N$
60 GOTO 20
70 END
```

ونجد أن البيانات المخزنة داخل N $ هى كالتالى :

قائمة	N$
NASHID	NASHID
AHMAD	AHMAD
FAYEZ	FAYEZ
	LAST

وهذا مثال مبسط، والبحث فى القائمة عن اسم شخص معين ، «فايز» مثلا ، أمر سهل . ولكن لو كان عدد أسماء الموظفين فى دائرة معينة كبيراً حوالى ٥٠٠ مثلاً فالبحث عن اسم معين يتطلب جهداً كبيراً ، إلا إذا كانت قائمة الأسماء مرتبة حسب الأحرف الهجائية .

وفى معظم الأحيان تكون الأسماء المقروءة عشوائية الترتيب ، و يطلب من الحاسب فرزها ، وإذا أردنا تحقيق ذلك ، يجب أن تخزن الأسماء كلها فى الذاكرة ، ومن ثم يتم فرزها ، وهذا يتطلب مقارنة بين متغيرات حرفية .

إن المتغيرات الحرفية يتم ترميزها داخل الحاسب بأرقام / أرقام وحروف وفق أحد النظامين (الثمانى أو السداسى عشر) (مقارنة بالنظام العشرى الذى يستخدمه الإنسان) . ولايهمنا هنا التعرض لكيفية تمثيل البيانات وفق هذين النظامين ، بل المهم أن نوضح الأساس الذى تستند إليه المقارنة بين المتغيرات الحرفية .

فمقارنة A مع B أو AB مع AC تستند إلى قيم رقمية لها معنى محدد للحاسب . فالحرف A له رقم أصغر من رقم B وأرقام الأحرف المجمعة AB أصغر من AC ...وهكذا .

وأحد الأساليب المقترحة لفرز الأسماء الثلاثة هو أن تخزن الأسماء الثلاثة فى صناديق مختلفة N1$, N2$, N3$ ومن ثم تتم المقارنة فيما بينها باستخدام عبارات IF / THEN لغرض فرزها ، وهذا يمكن تحقيقه إذا كان عدد الأسماء ثلاثة أو أربعة . أما إذا زاد العدد على ذلك ، فإن متابعة تسلسل عبارات IF / THEN تكاد تكون مستحيلة على العقل البشرى . وأدى هذا إلى تطوير أسلوب فى البرمجة يساعد فى التحكم فى البيانات التابعة لشىء معين موصوف ، أسماء الموظفين فى هذا المثال البسيط ، فبدلا من استخدام أسماء متغيرات لصناديق مستقلة ، فإنه يمكن الإشارة إلى مجموعة صناديق ذات علاقة وإطلاق اسم متغير واحد عليها ، ومن ثم تعريف كل صندوق برقم معين .

ومن أمثلة الحياة العملية عن هذا الأسلوب :

- نكتب بريد منطقة ما ، وأرقام صناديق البريد التابعة لها .
- اسم فندق معين وأرقام الغرف فيه .
- اسم شارع معين وأرقام المنازل فيه .
- اسم منطقة جغرافية معينة والرمز البريدى لها .

أما قواعد البرمجة الخاصة بهذا الأسلوب فتتم كالتالى :

يعطى اسم لمجموعة الصناديق ، و يشار إلى كل صندوق برقم متسلسل من ١ إلى عدد الصناديق المطلوبة ـــ ١٠ مثلاً ـــ وذلك كالشكل التالى :

شكل المصفوفة ذات البعد الواحد

N$ ← اسم المتغير للمجموعة								رقم الصندوق : X	
1	2	3	4	5	6	7	8	9	10
NASHID	AHMED	FAYEZ	LAST						

و يشار إلى كل صندوق كالتالى :

N$ (x)

حيث x تشير إلى رقم الصندوق ، و يبدأ من ١ ، ٢ ، ٣ ...إلى ١٠ .

ففى المثال السابق ، نود تخزين الأسماء فى الصناديق كما هوموضح سابقاً فمن الممكن تبديل سطر 10 ليظهر كالتالى :

10 READ N$ (1), N$ (2), N$ (3)

ومع إلغاء سطر 20 وتعديل سطر 40 بإلغاء «LAST» من نهايته ، يمكن قراءة الأسماء الثلاثة وتخزينها فى الصناديق الثلاثة الأولى ذات الأرقام ١ ، ٢ ، ٣ من اسم المجموعة N$.

ولكن ازدياد عدد الأسماء يودى بهذه الطريقة ويجعلها عديمة الاستعمال ، لذلك لابد من وجود وسيلة أخرى . وهذه الوسيلة تتطلب استخدام متغير (عداد) لأرقام الصناديق ومن ثم تغير هذا العداد كلما تم تخزين اسم من البيانات ، و بعد التعديلات عليه يصبح البرنامج كالتالى :

```
5 LET X = 1
10 READ N$(X)
20 IF N$(X) = "last" THEN 70
30 PRINT N$(X)
40 DATA "nashid","ahmed","fayez","last"
45 LET X = X + 1
50 READ N$(X)
60 GOTO 20
70 END
```

استخدام تعليمة DIM : وتتطلب المصفوفة تعليمة معينة تحدد عدد الصناديق التى تريد من الحاسب أن يحجزها لكل اسم . وتتبع هذه التعليمة الشكل التالى :

شكل عبارة		DIM	
(10), ...YX (20) ↓ وهكذا ..... عدد الصناديق	YY ↓ اسم المتغير	DIM ↓ تعليمة تحديد أبعاد المصفوفة	xxx ↓ رقم السطر

تطوير برنامج باستخدام المصفوفات ذات البعد الواحد :

مثال (١١ـ١) :

الهدف : حساب الربح الصافى لكل يوم فى الأسبوع لبائع الجرائد ، إذا علم أن سعر بيع الجريدة ريالان وتكلفتها نصف ريال وثمن المسترجع منها ٢٥ هللة (ربع ريال) ، وحساب نسبة مبيعات كل يوم إلى المجموع الكلى للمبيعات.

أولاً ــ طرق الحل :

عند تحليل هذا الهدف ، يتبين أنه فى الإمكان تقسيمه إلى جزأين :

الأول ــ يقوم بحساب الصافى اليومى والأسبوعى للمبيعات .

الثانى ــ يقوم بحساب نسبة المبيعات اليومية إلى المجموع الكلى للمبيعات .

ويمكن اتباع أحد ثلاثة طرق لتحقيق هذا الهدف :

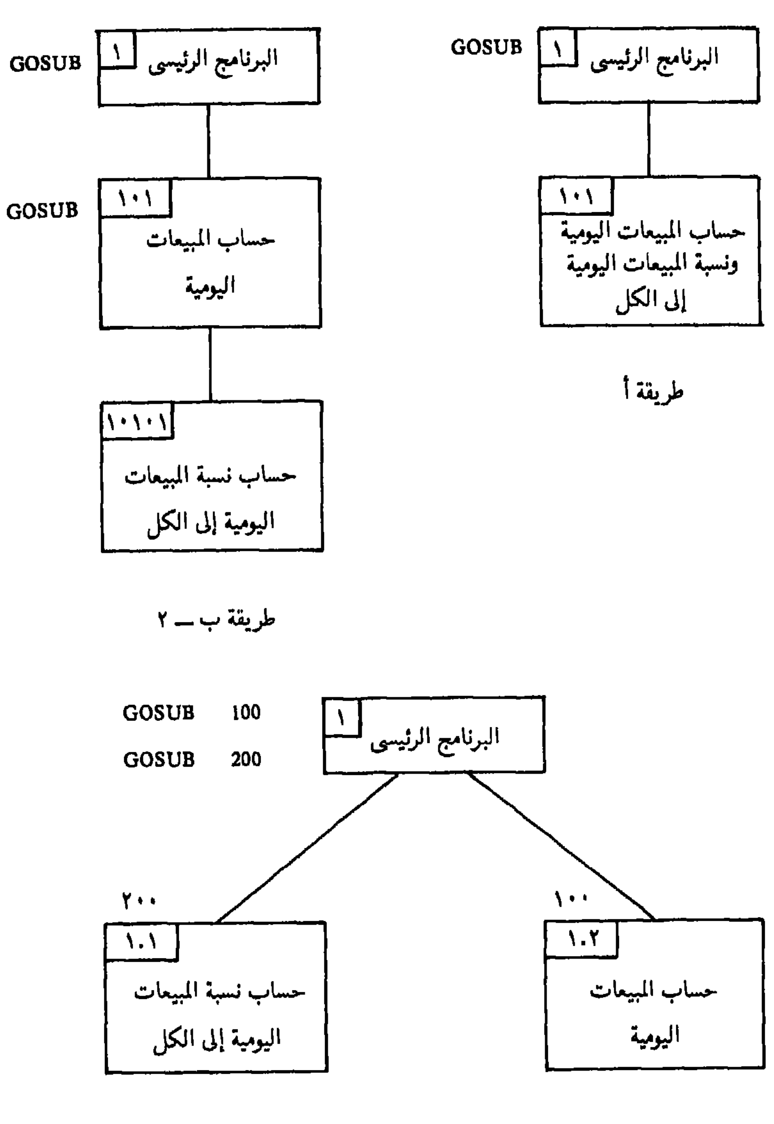
أ ــ أن يظهر الجزءان فى برنامج فرعى واحد .

ب ــ أن يظهر الجزءان فى برنامجين فرعيين مستقلين ، وفى هذه الحالة :

١ ــ إما أن يكون التنفيذ من البرنامج الرئيسى لكلا الجزأين بشكل متتالٍ .

٢ ــ وإما أن يكون تسلسل التنفيذ من البرنامج الرئيسى إلى البرنامج الفرعى الأول ومنه إلى البرنامج الفرعى الثانى .

ثانياً ـ الهياكل الهرمية حسب هذه الطرق الثلاثة مع تبيان الأجزاء التى ستظهر فيها تعليمة GOSUB للتوضيح .



ففى الشكل ب ــ ٢ نجد أن الجزء رقم ١٠١ يتحكم فى الجزء ١٠١٠١ ، وفى نفس الوقت يقوم بالمعالجات الخاصة به ، ففى هذه الحالة يفضل ألا تكون تبعية الجزء ١٠١٠١ له ؛ وذلك للأسباب التالية :

١ ــ أن تعديل الجزء ١٠١ قد يؤدى إلى التأثير على الجزء ١٠١ .

٢ ــ أن وظيفة الجزء المتحكم الرئيسية هى توجيه التنفيذ للأجزاء الأخرى التابعة له .

٣ ــ صعوبة تتبع خطوات البرنامج ، وخاصة إذا تكرر هذا النوع من العلاقة بين الجزء المتحكم والمنفذ فى برنامج ما .

وقد تم استخدام الأسلوب الأول لتطوير البرامج للأمثلة المشروحة فى الفصول السابقة ؛ لذلك فسنقوم بتطوير البرنامج حسب الهيكل الهرمى ب ــ ١ .

الخطوات الرئيسية فى الجزء الأول :

١ ــ لكل يوم من أيام الأسبوع ، اقرأ الكمية المشتراة ، والكمية المبيعة .

٢ ــ لكل يوم احسب صافى الربح = الكمية المبيعة * 2.0 ــ الكمية المشتراة * 50 . + (الكمية المشتراة ــ الكمية المبيعة) * 25 .

٣ ــ اطبع الناتج .

٤ ــ احسب المبلغ الإجمالى المتجمع بعد كل يوم .

٥ ــ وبعد انتهاء معالجة اليوم السابع ، اطبع المبالغ الكلية ثم توقف .

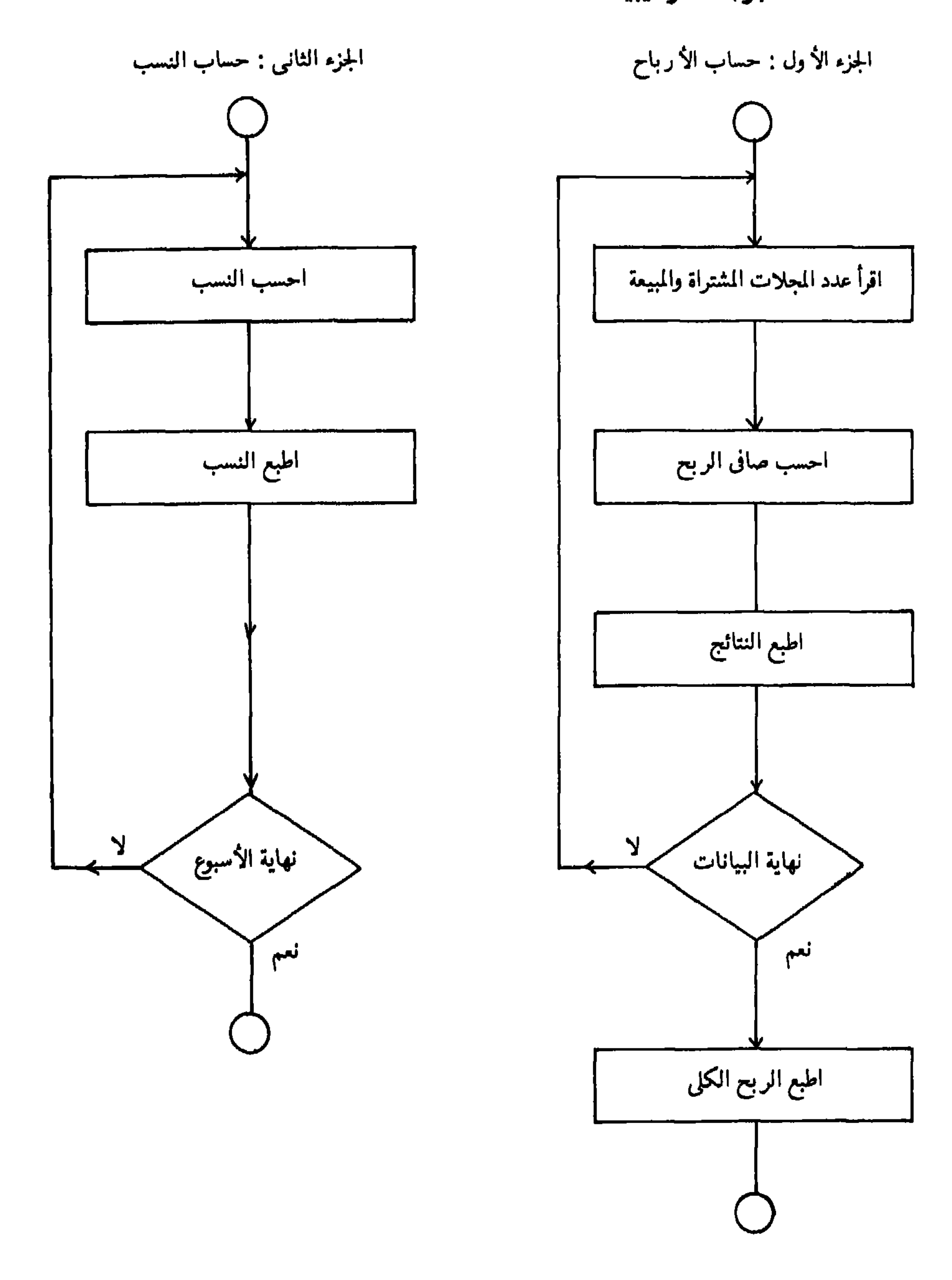
خطوات الحل للجزء الثانى :

١) لكل يوم من أيام الأسبوع ، احسب نسبة الربح إلى المجموع الكلى للربح .

٢) اطبع الناتج .

٣) بعد طباعة آخر نسبة ، توقف .

ثالثاً ــ نمط البرمجة التركيبية :



رابعاً ــ تطوير البرنامج :

وكيفية تحقيق ذلك باستخدام المصفوفات تتم حسب الجدول التالى :

C (I) النسبة من كل الأرباح	R (I) صافى الربح	S (I) العدد المبيع	P (I) عدد المشترى	اسم المصفوفة رقم الأسبوع M
XX.XX%	XX.XX	40	50	1
XX.XX	XX.XX	15	30	2
XX.XX	XX.XX	45	60	3
XX.XX	XX.XX	60	80	4
XX.XX	XX.XX	40	70	5
XX.XX	XX.XX	15	35	6
XX.XX	XX.XX	45	55	7
المجموع	XXXXXX	T ⟶		↑ رقم الصندوق فى كل مصفوفة

من الجدول السابق يتبين أننا بحاجة إلى أربع مصفوفات ذات بعد واحد ، ومتغير عداد لمعرفة أرقام الأسبوع ، ومتغير جامع لمعرفة مجموع المبيعات الكلية . لاحظ أن العداد M يرمز إلى رقم الصندوق لكل مصفوفة بالإضافة إلى رقم الأسبوع، ويمثل الشكل التالى البرنامج الكامل حسب الهيكل الهرمى ب ــ ١ .

شكل (١١ ـ ١)

برنامج لحساب صافى أرباح بائع الجرائد أسبوعياً ونسبة ربح كل يوم إلى مجموع الأرباح

```
10 REM
20 GOSUB 50 : REM    استدعاء برنامج فرعي لحساب الا رباح و مجموعها
30 GOSUB 160 : REM   استدعاء برنامج فرعي لحساب نسبة الا رباح من  المجموع
40 END
50 REM   برنامج فرعي لحساب الا رباح
60 LET M = 1
70 REM  لقراءة الا عداد المشتراه  و  المباعه
80 READ P(M),S(M)
90 REM  لحساب  صافي الا رباح
100 LET R(M) = S(M) * 2! - P(M) * .5 + (P(M) - S(M)) * .25
110 LET T = T + R(M)
120 LET M =  M + 1
130 IF M <= 7 THEN 80
140 DATA 50,40,30,15,60,45,80,60,70,40,35,15,55,45
150 RETURN
160 REM  برنامج فرعي لحساب النسب و طباعة النتائج
170 PRINT "        رقم          العدد            العدد     صافي الربح        "النسبه من
180 PRINT " الا سبو ع       المشترى          المبا ع                        "كل الا رباح
190 PRINT "=========    ========       ========   ===========     =========="
200 LET W = 1
210 LET C(W)=(R(W) / T) * 100
220 LET P$=  "   ##               ###              ###        ###.##           ##.##"
230 PRINT USING P$;W,P(W),S(W),R(W),C(W)
240 LET W = W + 1
250 REM  للا ستفسار عن نهاية الا سابيع
260 IF W < = 7 THEN 210
270 PRINT "=================================================================="
280 PRINT TAB(42);T;" =  مجموع الا رباح"
290 PRINT "=================================================================="
300 RETURN
```

وعند تنفيذ البرنامج ، سنحصل على النتائج التالية :

رقم الا سبو ع	العدد المشترى	العدد المبا ع	صافي الربح	النسبه من كل الا رباح
1	50	40	57.50	15.97
2	30	15	18.75	5.21
3	60	45	63.75	17.71
4	80	60	85.00	23.61
5	70	40	52.50	14.58
6	35	15	17.50	4.86
7	55	45	65.00	18.06

360 = مجموع الا رباح

لاحظ أن المتغير العداد M و W قد استخدم للدلالة على رقم الأسبوع ، بالإضافة إلى رقم الصندوق فى المصفوفة . لاحظ أيضا أن تعليمة DIM لم تظهر فى هذا البرنامج ، إذ أن البرنامج يحدد تلقائياً ١٠ صناديق لكل متغير استعمل كمصفوفة . ولكن من الأفضل إضافتها دائما إلى البرنامج ؛ منعاً للإرباك والمزج بين متغيرات المصفوفات والمتغيرات العادية ، و يتم ذلك فى البرنامج السابق كالتالى :

```
107     DIM     R (10) , S(10), P(10)
505     DIM     C (10)
```

مقدمة عن الدوارة البسيطة :

تحدثنا فى المثال السابق عن استخدام المتغير العداد ؛ للتحكم فى عمليات إدخال وطبع البيانات إلى ومن المصفوفات ذات البعد الواحد . ونظراً لتعدد هذا النوع من التطبيقات فى المجالين التجارى والحكومى ، كان لابد من تطوير تعليمات برمجية خاصة لتسهيل هذا النوع من المعالجات .

ولاحظنا أن هناك ثلاث حالات للمتغير العداد «I» فى المثال السابق ، يجب مراعاتها :

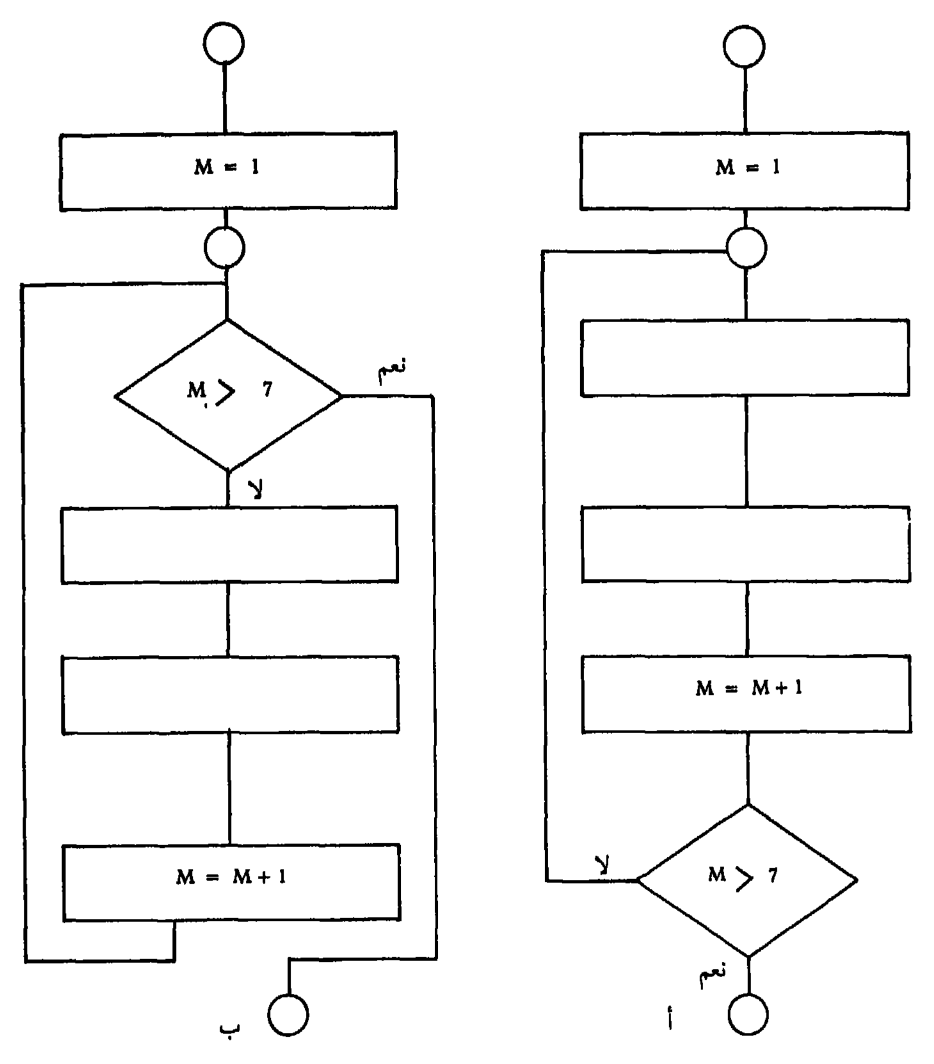
١ ــ القيمة الأ ولية ، وكانت ١

٢ ــ القيمة النهائية ، وكانت ٧

٣ ــ مقدار الزيادة فى العداد وكانت ١

وكان لابد من تحديد هذه القيم الثلاث للعداد المستخدم فى التحكم فى الدوارة ، سواء كان ذلك عند استخدام المصفوفة أو عند التحكم فى تنفيذ مجموعة عبارات بعضها ذات علاقة مع بعض .

و يبين الرسمان التاليان تسلسل العمليات المنفذة للتحكم فى الدوراة بأسلوبين مختلفين وفق أساليب البرمجة التركيبية .



وكما مر ذكره فى الفصل السابع ، لوحظ أن الرسم أ هو الذى استخدم فى البرنامج السابق شكل (١١ ــ ١) ، وذلك للتقليل من عدد استخدامات تعليمة GO TO . ولكن الرسمين سليمان من وجهة نظر منطقية . وسنتحدث الآن عن التعليمات التى تسهل عملية التحكم فى الدوارة البسيطة ، حيث تستخدم هذه التعليمات المنطق الموضح فى الشكل ــ ب ــ السابق.

تطوير برنامج باستخدام تعليمتى FOR / NEXT للتحكم فى الدوارة البسيطة :

فى المثال التالى ، سيتم تطوير برنامج باستخدام تعليمات الدوارة البسيطة ؛ للتحكم فى تنفيذ عبارات ذات علاقة ، ودون استخدام المصفوفات ، وذلك باستخدام مثال محاسبى .

مثال (١١ ـ ٢) :

الهدف : حساب قيمة الاستهلاك للأصول الثابتة بطريقة الاستهلاك المتناقص ، إذا علمت القيمة الأصلية الدفترية وعدد سنوات الاستخدام ، وذلك بأسلوب مجموع أعداد السنوات .

أولاً ـ خطوات الحل :

١ ـ الحصول على القيمة الشرائية وسنوات الاستخدام .

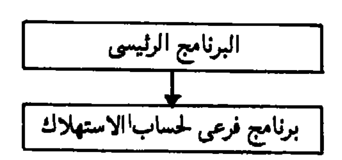
٢ ـ حساب مجموع أرقام السنوات ، مثال :

إذا كان الاستهلاك لمدة ٥ سنوات ، يكون مجموع أرقام السنوات : ١+٢+٣+٤+٥ = ١٥ و يكون مقدار الاستهلاك السنوى للأصل = القيمة الدفترية* رقم السنة ÷ ١٥ ، وذلك بدءاً بالرقم ٥ للسنة الأولى وانتهاء بالرقم ١ للسنة الخامسة .

٣ ـ حساب الاستهلاك السنوى لكل السنوات .

ثانياً ـ الهيكل الهرمى : يمكن تطوير الهيكل الهرمى بأحد أسلوبين :

أ) وجود جزء واحد ، وذلك حسب الشكل التالى ·



ب ـ وإذا أريد استخدام أكثر من جزء واحد ، فيمكن الأخذ بعين الاعتبار للوظائف التالية :

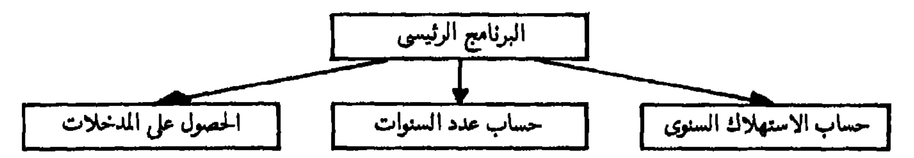
١ ـ الحصول على المدخلات .

٢ ـ حساب عدد السنوات .

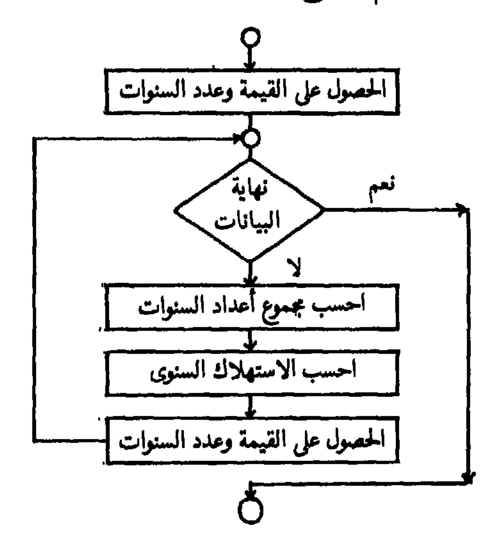
٣ ـ حساب الاستهلاك السنوى لكل السنوات مع الطباعة للنتائج .

حيث يتم تدقيق نهاية البيانات عن طريق البرنامج الرئيسى ، و يرجع سبب ذلك إلى ضرورة إنهاء البرنامج فى الجزء الرئيسى وليس فى أى جزء فرعى .

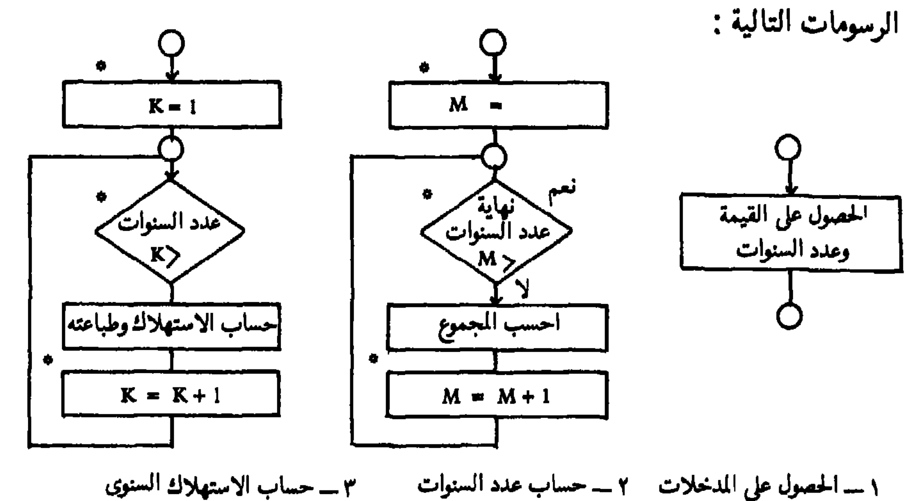
و يعكس الرسم التالى الهيكل الهرمى الذى يحوى ثلاثة أجزاء فرعية .



ثالثاً : رسم البرمجة التركيبية : يعكس الرسم التالى الخطوات المتبعة لتحقيق الهدف من المثال ، على فرض استخدام برنامج جزئى واحد .



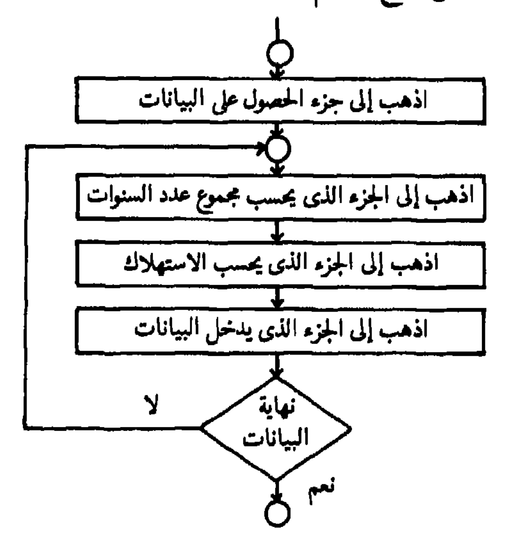
ووفق الأسلوب ب ، يمكن وضع الوصف التفصيلى للخطوات عن طريق الرسومات التالية :



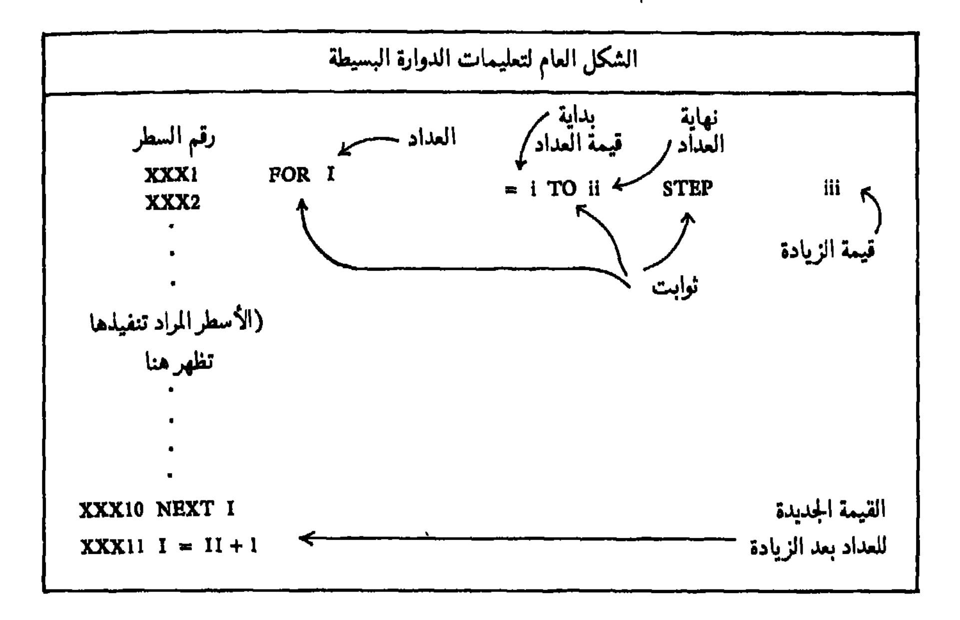
١ ــ الحصول على المدخلات   ٢ ــ حساب عدد السنوات   ٣ ــ حساب الاستهلاك السنوى

لاحظ الرموز المؤشر عليها بـ «*» والتى تشير إلى الخطوات الأساسية للتحكم فى الدوارة .

ولاحظ أن البرنامج الرئيسى يتبع الرسم التالى للبرمجة التركيبية :



و يتطلب الجزء الثانى والثالث أسلوباً للتحكم فى الدوارة يمكن تنفيذه باستخدام تعليمتى NEXT / FOR ، كما هو موضح فى الشكل التالى :



حيث تحصر العبارات المراد التحكم فى تنفيذها بين عبارتى FOR و NEXT .

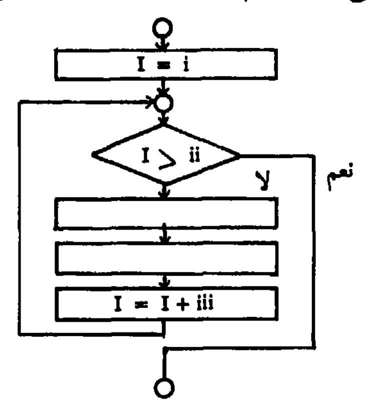
وتقوم تعليمة FOR بالعمليات التالية :

١ ــ إسناد القيمة الأولية (بداية العداد) مرة واحدة للمتغير العداد ــ I الشكل السابق : I = i .

٢ ــ التدقيق فيما إذا كانت قيمة I تزيد على القيمة النهائية للعداد ، فى هذه الحالة I > ii فإذا كانت قيمة العداد تزيد على القيمة المعطاة يتم التشعب إلى السطر الذى يلى عبارة NEXT ، XXX 11 ، وإلا يستمر التنفيذ إلى السطر الذى يلى عبارة FOR ، XXX 2 .

أما تعليمة NEXT فتحسب القيمة الجديدة للمتغير العداد حسب ماتتطلبه القيمة المحددة فى عبارة FOR وعلى وجه التحديد القيمة الموجودة بعد كلمة STEP ,iii .

لذلك نرى أن عبارتى FOR / NEXT تتحكمان فى الدوارة على الشكل التالى :



ويمكن تتبع مايحدث لقيمة I وكيفية المقارنة والتشعب حسب الجدول التالى :

التعليمة	قيمة	المقارنة	الحالة	نتيجة المقارنة
FOR	I = i	I > ii	لا	استمر
NEXT	I = I + iii			
FOR		I > ii	لا	استمر
NEXT	I = I + iii			
FOR		I > ii	لا	استمر
NEXT	I = I + iii			
FOR		I > ii	نعم	توقف

وسنعطى مثالاً بسيطاً لمراقبة مايحدث لقيمة I ، كما فى البرنامج التالى :

```
10 FOR I = 1 to 4 STEP 1
20 PRINT I
30 NEXT I
40 PRINT «القيمة خارج الدوارة» ; I
```

وعند تنفيذ البرنامج تكون المخرجات كالتالى :

```
1
2
3
4
5 القيمة خارج الدوارة
```

رابعاً ــ البرنامج كاملاً فى الشكل التالى :

شكل (١١ ــ ٢)

برنامج لحساب قيمة الاستهلاك المتناقصة وفق أسلوب عدد أرقام السنوات

```
10 REM برنامج رئيسي ويتم تدقيق نهاية البيانات فيه
20 GOSUB 90  ' للحصول على المدخلات
30 IF V = -999 THEN 80' تدقيق نهاية البيانات
40 GOSUB 120  ' عدد السنوات
50 GOSUB 180  ' لحساب الاستهلاك السنوي
60 GOSUB 90  ' على المدخلات
70 IF V <> -999 THEN 40' تدقيق نهاية البيانات
80 END
90 REM 'برنامج فرعي لادخال قيمة الاصل وعدد السنوات
100 INPUT"ادخل قيمة الاصل وعدد سنوات الاستهلاك او -٩٩٩ للانتهاء";V,Y
110 RETURN
120 REM برنامج فرعي لحساب مجموع السنوات
130 LET II= 0
140 FOR I = 1 TO Y STEP 1
150 LET II = II + I
160 NEXT I
170 RETURN
180 REM لحساب ما استهلك في السنه لكل السنوات
190 FOR I = 1 TO Y STEP 1
200 LET D = V *(6-I)  / II
210 PRINT I;" ما استهلك في السنه رقم ";D
220 NEXT I
230 RETURN
```

وفى حالة تنفيذ البرنامج، سنحصل على المخرجات التالية :

أدخل قيمة الأصل وعدد سنوات الاستهلاك أوــ ٩٩٩ للانتهاء ? 1000,2

1 666.6667

2 333.3334

أدخل قيمة الأصل وعدد سنوات الاستهلاك أوــ ٩٩٩ للانتهاء ? 500,3

1 250

2 166.6667

3 83.33334

أدخل قيمة الأصل وعدد سنوات الاستهلاك أوــ ٩٩٩ للانتهاء ? 999,0

OK

ملاحظات على البرنامج :

١ ــ استخدام العداد I للتحكم فى الدوارة فى كلا الجزأين ، وهذا لايؤثر مطلقاً على سير البرنامج ؛ إذ تبدأ القيم الجديدة لها فى الجزء الثانى بعد الانتهاء من الجزء الأول .

٢ ــ عند الانتهاء من تنفيذ الدوارة ، تكون قيمة العداد I أكبر من القيمة النهائية بمقدار الزيادة المحددة عقب STEP BY .

٣ ــ يحوى السطر 210 أسلوباً مبسطاً للحصول على القيم العكسية للعداد I وذلك حسب الجدول التالى :

قيمة I	القيمة المطلوب استخدامها فى المعادلة	كيفية الحصول عليها
1	5	6 - 1
2	4	6 - 2
3	3	6 - 3
4	2	6 - 4
5	1	6 - 5

وهذا يعنى أنه من الممكن استخدام عداد الدوارة فى معادلة جبرية داخل الدوارة المنفذة ، شريطة ألا تظهر فى الجهة اليسرى لإشارة = ، إذ أن تغير قيمة عداد الدوارة داخل العبارة المنفذة يؤدى إلى تغيير عدد المرات التى ستنفذ معها هذه العبارة (المحصورة بين تعليمتى FOR و NEXT) .

ففى المثال التالى ، نجد أن وضع عداد الدوارة إلى يمين إشارة = فى معادلة جبرية تقع ضمن العبارات الواقعة بين FOR و NEXT ، أدى إلى تغيير عدد المرات التى تنفذ بها هذه العبارة :

<table>
<tr><th colspan="2">تعديل قيمة عداد الدوارة فى عبارة بين FOR و NEXT</th></tr>
<tr><td>FOR / NEXT
بشكلها الصحيح</td><td>FOR / NEXT
تعديل قيمة I عداد الدوارة</td></tr>
<tr><td>10 FOR I = 1 TO 5 STEP BY 1

100 NEXT I</td><td>10 FOR I = 1 TO 5 STEP BY 1
50 I = I + 2
100 NEXT I</td></tr>
<tr><td>أ</td><td>ب</td></tr>
</table>

ففى الشكل ــ أ ــ يتم تنفيذ العبارات خمس مرات ، أما فى الشكل ــ ب ــ فيتم تنفيذ العبارات مرتين فقط ، وشرح ذلك كالتالى :

نتيجة المقارنة	الحالة	المقارنة	قيمة	رقم السطر
استمر	لا	1 > 5	1	10
		لا توجد مقارنة بعد	3	50
استمر	لا	4 > 5	4	100
		لا توجد مقارنة بعد	6	50
توقف	نعم	7 > 5	7	100

٤ ــ التسلسل المتناقص للعداد : فى الفقرة السابقة تم شرح كيفية الحصول على القيم المتنزلة لعداد الدوارة باستخدام قيمها المتصاعدة ، فى حالات كهذه يمكن أن تظهر القيم المتحكمة فى عداد الدوارة بشكل متناقص ، ففى المثال السابق يظهر السطر 150 كالتالى :

```
150 FOR I = 5 TO 1 STEP — ١
```

و يتم تعديل السطر 150 ليتماشى مع الشكل الجديد لعبارة FOR

```
150 LET D = V * I / II
```

بالإضافة إلى إمكانية استخدام الأرقام السالبة فى عبارة FOR ، يمكن استخدام قيم عشرية للمتغيرات فالعبارات التالية كلها صحيحة :

```
FOR  I = 1 TO 10 STEP .1
FOR  J = 1 TO 1.5 STEP .01
FOR  M = 3.6 TO 0 STEP -.1
```

٥ ــ إمكانية استخدام متغيرات بدلاً من قيم فى عبارة FOR فالمتغير Y والذى يمثل عدد السنوات ، استخدم فى السطرين 150 ,200 للدلالة على القيمة النهائية المطلوبة لعداد الدوارة . وكذلك يمكن أن يكون أى من القيم أو جميعها ممثلة فى متغيرات ، بشرط إسناد قيم لها قبل تنفيذ عبارة FOR .

فإذا أردنا مثلاً تحديد بداية السنة المطلوب فيها الاستهلاك ، يمكن إدخال هذه القيمة مع بقية المدخلات ، و يتم تغيير البرنامج كالتالى :

V, Y, Y1 ; «أدخل القيمة للأصل مع سنوات الاستهلاك و بداية السنوات أو 999 للانتهاء» 110 INPUT

```
150 FOR I = Y1 TO Y STEP 1
20 FOR I = Y1 TO Y STEP 1
```

وكذلك الأمر إذا كانت الزيادة لأكثر من سنة واحدة ، أى حساب قيمة الاستهلاك كل سنتين ، فإذا أضفنا المنفذ Y2 فى نهاية السطر 110 فتكون عبارة FOR كالتالى :

```
110 FOR I = Y1 TO Y STEP Y2
```

٦ ــ استخدام تعابير جبرية للتحكم فى قيم عداد الدوارة :

إذا أردنا حساب الاستهلاك نصف السنوى عوضاً عن الاستهلاك السنوى

للأصل ، تصبح الأرقام الخاصة بالحساب ضعف القيمة المدخلة للسنوات ، أى ١٠ بدلاً من ٥ مثلا .

ويمكن تعديل عبارتى FOR لحساب الاستهلاك نصف السنوى كالتالى :

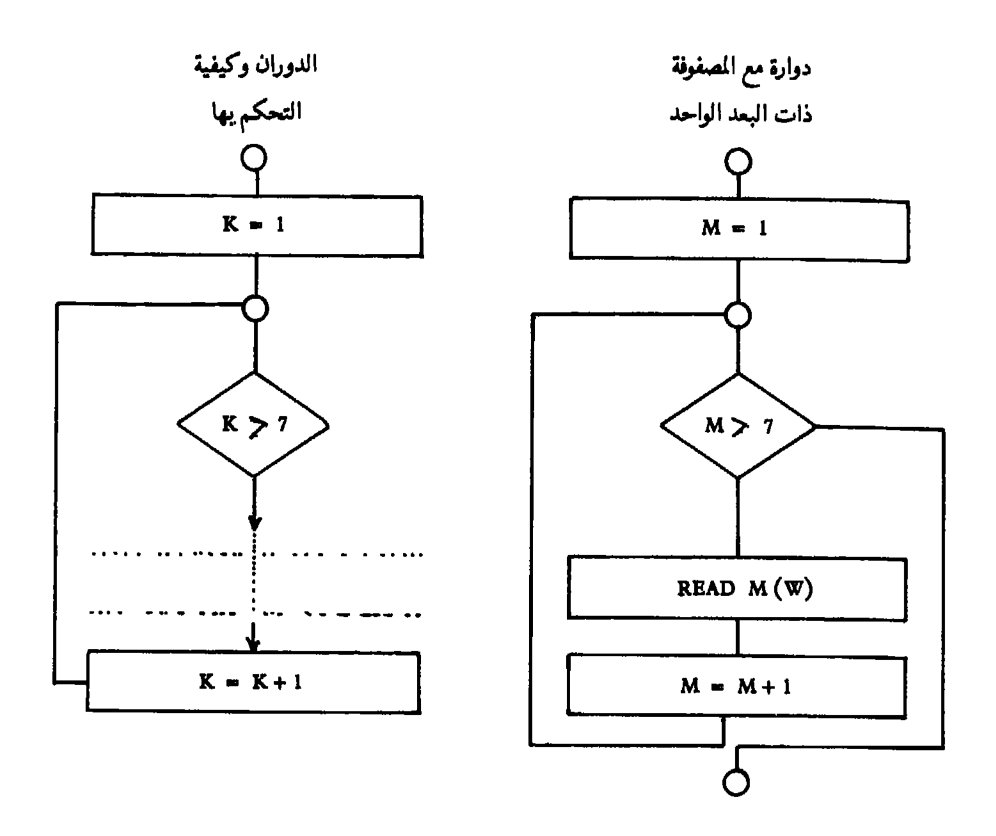
```
150  FOR  I  =  1  TO  Y * 2  STEP  1
200  FOR  I  =  1  TO  Y * 2  STEP  1
```

وكذلك من الممكن أن تحل العبارات الجبرية محل أى من المتغيرات أو القيم العددية فى عبارة FOR فالأشكال التالية لعبارة FOR صحيحة :

```
FOR  I  =  (I1 / K1)  TO  W2 * 3  STEP  Y / Z
FOR  I  =  (I1 + I2) / I3  TO  I4 * I5 / I6  STEP
           (I6 + I7) / I8
FOR  I  =  1  TO  Y * Y2 / 2  STEP        Y3 / 3
```

تطوير برنامج باستخدام FOR / NEXT لمعالجة المصفوفات ذات البعد الواحد :

تبين مما سبق أن عبارتى FOR / NEXT تسهلان عملية التحكم فى معالجة مجموعة من العبارات ذات العلاقة فيما بينها ، وكيف أن التخزين فى المصفوفة ذات البعد الواحد يتطلب استخدام رقم متسلسل للصندوق ، وأن التحكم فى هذا الرقم يخضع لنفس الأسس التى يخضع لها التحكم فى الدوارة ، و يوضح ذلك الشكلان التاليان :



فعند استخدام تعليمتى FOR / NEXT مع المصفوفة ذات البعد الواحد ، يتم فى العادة استخدام عداد الدوارة كرقم للصندوق الذى يظهر بين قوسين بعد اسم المصفوفة .

ففى المثال الأول فى هذا الفصل ، يتم تعديل البرنامج الموضح فى شكل ١ عند استخدام FOR / NEXT كالتالى :

الأسطر التالية تتحكم فى الدوارة الأولى والثانية :

```
105   LET  M = 1
140   LET  M = M+1
150   IF   M <= 7 THEN 110
```

```
500   LET  W = 1
```
برنامج فرعى لحساب النسب وطباعة النتائج

```
525   LET  W = W + 1
580   IF   W <= 7 THEN 510
```

يتم استبدال الأسطر التالية بعبارات FOR / NEXT

```
105   FOR   M = 1 TO 7
150   NEXT  M
```

برنامج فرعى لحساب النسب وطباعة النتائج

```
500   FOR   W = 1 TO 7
530   NEXT  W
```

ومع إزالة السطرين 140, 525 يصبح البرنامج كاملاً وصحيحاً و يؤدى نفس النتائج . وتنطبق هنا نفس الملاحظة التى ظهرت عقب شكل ١١ ـــ ١ مع الفارق أن عداد الدوارة هو الذى استخدم كرقم الأسبوع ورقم الصندوق ، فى المصفوفات المختلفة بدلاً من أن يكون المتغير العداد .

وسنتعرض الآن لتطوير برنامج آخر باستخدام FOR / NEXT والمصفوفة ذات البعد الواحد .

مثال (١١ ـــ ٣) :

الهدف : قراءة أعمار الموظفين وتقسيمها حسب الفئات التالية ، ومن ثم طبع هذه الأعمار الثابتة لكل فئة مع عددها ونسبة كل فئة :

الفئة الأ ولى : أقل من ٢٠ سنة .

الفئة الثانية : ٢٠ إلى أقل من ٣٠ سنة .

الفئة الثالثة : ٣٠ إلى أقل من ٤٠ سنة .

الفئة الرابعة : ٤٠ فأكثر .

أولاً ــ خطوات الحل :

أ) الحصول على العمر حتى نهاية البيانات .

ب) معرفة الفئة الخاصة به وتخزين العمر فى مصفوفة .

جـ) عند الانتهاء من التصنيف ، طباعة الفئات وإلى جانبها الأعمار .

وبحيث يكون شكل المخرج كالتالى :

الأعمار	النسبة	العدد	الفئة
xxxx	xx/xx	xx	أقل من ٢٠ سنة
xxxx	xx/xx	xx	٢٠ سنة إلى أقل من ٣٠
xxxx	xx/xx	xx	٣٠ سنة إلى أقل من ٤٠
xxxx	xx/xx	xx	٤٠ سنة فأكثر

وإذا افترضنا وجود الأعمار التالية كمثال :

١٩ ، ٢٢ ، ٤٥ ، ٣١ ، ٣٠ ، ٢٠ ، ٤٠ ، ١٨ ، ٤٩ ، ٣٩

فسيكون شكل المخرج كالتالى :

الأعمار	النسبة	العدد	الفئة
19 18	20٪	2	أقل من ٢٠ سنة
22 20	20٪	2	٢٠ سنة إلى أقل من ٣٠
31 30 39	30٪	3	٣٠ سنة إلى أقل من ٤٠
45 40 49	30٪	3	٤٠ سنة فأكثر
	100٪	10	المجموع

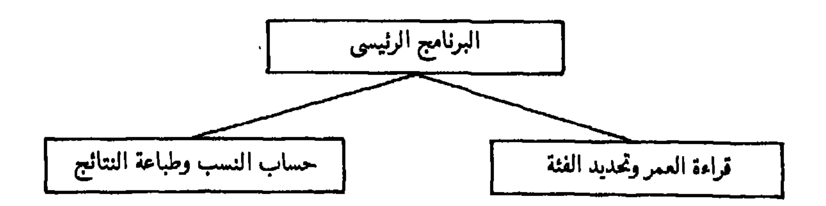
ثانياً ــ رسم الهيكل الهرمى :

يمكن أن يقسم البرنامج إلى الأجزاء التالية :

أ) قراءة العمر وتحديد الفئة وتخزين العمر .

ب) حساب النسب وطباعة المخرجات .

و يعكس الرسم التالى الهيكل الهرمى لهذا المثال :



ثالثاً ــ أشكال البرمجة التركيبية :

نهاية البيانات

قراءة العمر

الجزء الأول :

الفئة الأولى — نعم — تخزين العمر والعدد

لا

الفئة الثانية — نعم — تخزين العمر والعدد

لا

الفئة الثالثة — نعم — تخزين العمر والعدد

لا

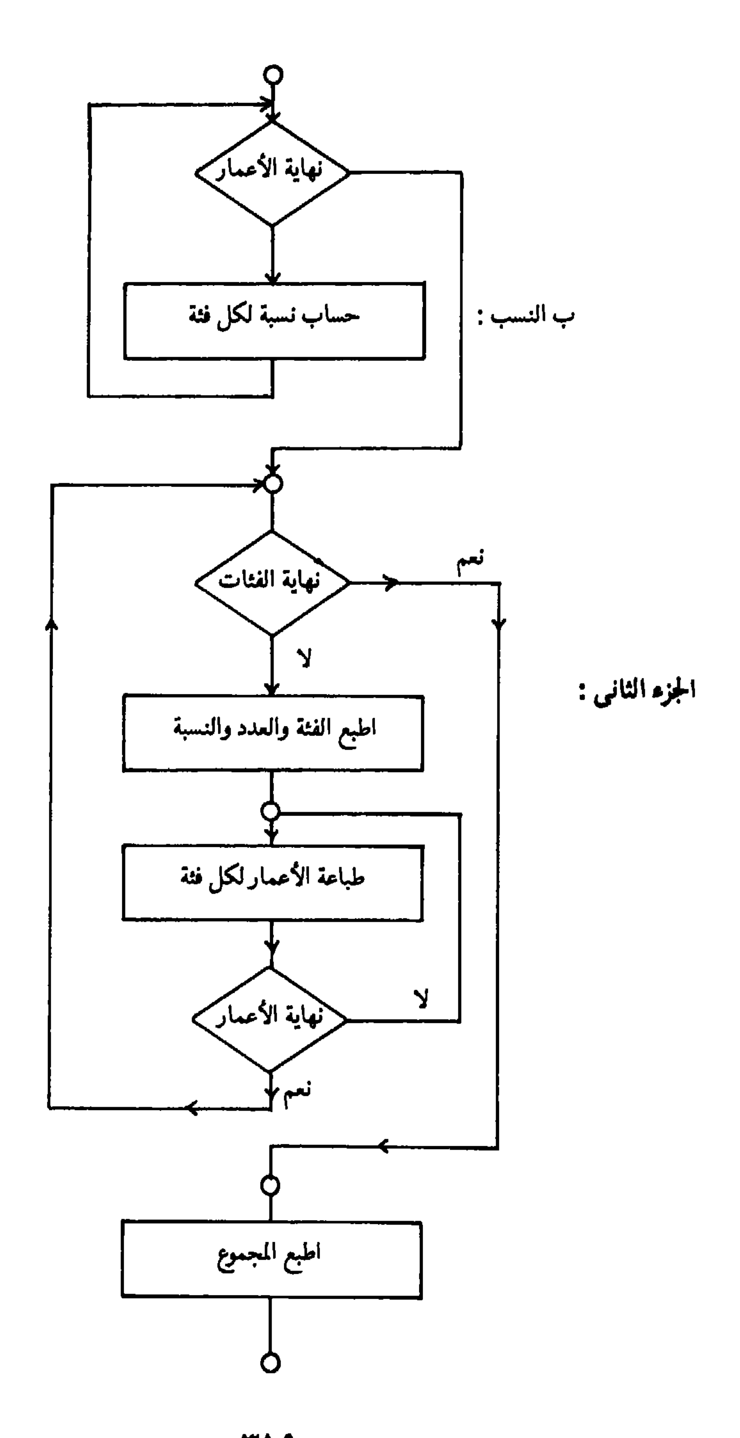
الفئة الرابعة — نعم — تخزين العمر والعدد

لا

احسب مجموع أعداد الأعمار

ب النسب :

الجزء الثانى :

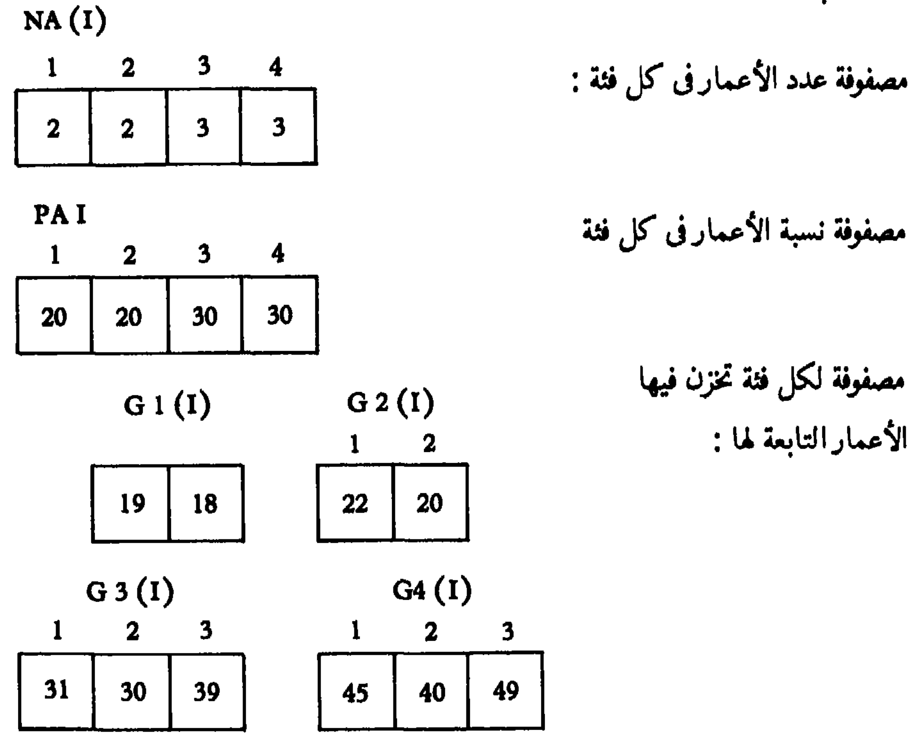


ملاحظات على أشكال البرمجة التركيبية : نلاحظ من الأشكال السابقة ، التالى :

١ ـ فى الجزء الأول ، نجد أن مصفوفة لكل فئة قد استخدمت وذلك لتخزين الأعمار فى كل منها ، كما نلاحظ وجود عداد متغير لكل فئة فى مصفوفة واحدة .

٢ ـ فى الجزء الثانى ، نجد أن طباعة المخرجات تمت عن طريق تنفيذ دوارتين متداخلتين : الأولى لطباعة الفئات الأربع واحدة واحدة ، والثانية لطباعة الأعمار التابعة لكل فئة ، وسيتم التحكم فى الدوارة الأولى بالطريقة العادية وفى الثانية عن طريق FOR / NEXT .

٣ ـ يكون مجموع المصفوفات المطلوبة كالتالى مطبقاً على مثال الأعمار المشار إليه سابقاً :



رابعاً ـ البرنامج كاملاً فى الشكل التالى :

شكل (١١ ـ ٣)

برنامج لتصنيف الأعمار إلى فئات أربع، وطباعة أعمار كل فئة مع عددها ونسبتها.

```
10 REM  برنامج رئيسي لقراءة الاعمار و تصنيفها و تعدادها
20 DIM NA(4),PA(4),G1(10),G2(10),G3(10),G4(10)
30 GOSUB 100: REM  استدعاء برنامج فرعي لقراءة الاعمار و تصنيفها و تعدادها
40 GOSUB 500: REM  استدعاء برنامج فرعي لحساب النسب و طباعة النتائج
50 END
100 REM  برنامج فرعي لقراءة الاعمار و تصنيفها و تعدادها
105 REM  اسناد القيم الاوليه للمتغيرات
107 LET C1=0: C2=0: C3=0: C4=0
109 REM  دواره بعدد القيم
110 FOR I = 1 TO 10
120 READ A: REM  لقراءة العمر
125 REM  لتصنيف الاعمار حسب قيمها
127 REM  الصنف الاول
130 IF A < 20 THEN C1 = C1 + 1: G1(C1)=A:NA(1)=C1
135 REM  الصنف الثاني
140 IF A >= 20 AND A < 30 THEN C2=C2+1: G2(C2)=A:NA(2)=C2
145 REM  الصنف الثالث
150 IF A >= 30 AND A < 40 THEN C3=C3+1: G3(C3)=A:NA(3)=C3
155 REM  الصنف الرابع
160 IF A >= 40  THEN C4=C4+1: G4(C4)=A:NA(4)=C4
170 NEXT I
180 LET T = C1 + C2 + C3 + C4
185 DATA 13,24,42,64,31,19,20,11,27,38
190 RETURN
500 REM  لحساب النسب و طباعة النتائج
505 REM  دواره بعدد النسب
510 FOR K=1 TO 4
520 LET PA(K) = (NA(K) / T) * 100  : REM  لايجاد النسبه المئويه
530 NEXT K
540 REM  لطباعة النتائج باستخدام دوارتين متداخلتين
550 REM  اسناد العناوين لمتغيرات
560 LET A1$(1) = "اقل من ٢٠ سنه": LET A1$(2) = "٢٠ سنه و اقل من ٣٠"
570 LET A1$(3) = "٣٠ سنه و اقل من ٤٠": LET A1$(4) = "اكثر من ٤٠ سنه"
580 REM  بدء الطباعه
590 LET I=1
600 IF I > 4 THEN 710
610 PRINT " % ";PA(I);" = النسبه     ";
675 PRINT "          (";NA(I);")    ";A1$(I)
680 PRINT " "
690 LET I = I + 1
700 GOTO 600
710 RETURN
```

وفى حالة تنفيذ البرنامج شكل (١١ ـ ٣) سنحصل على النتائج التالية :

```
%  30  =  النسبه          ( 3 )     اقل من ٢٠ سنه
%  30  =  النسبه          ( 3 )     ٢٠ سنه و اقل من ٣٠
%  20  =  النسبه          ( 2 )     ٣٠ سنه و اقل من ٤٠
%  20  =  النسبه          ( 2 )     اكثر من ٤٠ سنه
```

ملاحظات على البرنامج :

١ ــ لاحظ وجود ; فى نهاية تعليمة PRINT وهذا يؤدى إلى استمرار الكتابة على نفس السطر عند تنفيذ تعليمة PRINT التالية . لذلك كانت الحاجة إلى سطر 670 لكى نبدأ بالطباعة على سطر جديد عند الانتهاء من كتابة الأعمار لكل فئة ، وإلا بدأ بطباعة الكلمات الواصفة لكل فئة عقب الانتهاء من طباعة الأعمار ، ويمكنك إزالة السطر 670 ولتنفيذ البرنامج تراقب التغير فى طباعة النتائج .

٢ ــ استخدام المصفوفة (I) NA فى عبارة FOR السطر 610 . وهذا جائز أيضا ، وفى هذه الحالة ، تكون القيم إما ٢ ، ٢ ، ٣ أو ٣ .

٣ ــ لاحظ عدد المرات التى ستنفذبها الأسطر التالية وفق البيانات المعطاة :

عدد المرات	السطر
٢	610
٢	630
٣	640
٣	650

٤ ــ تنفيذ الدوارة الداخلية والدوارة الخارجية : لاحظ أن الدوارة الخارجية ستنفذ الأسطر من 590 إلى 690 أربع مرات ، ولكل مرة من هذه المرات سيتم تنفيذ الأسطر 620 ، 630 ، 640 أو 650 ، بعدد الأعمار لكل فئة كما هو مبين سابقاً .

و يوضح ذلك الجدول التالى :

NA (I)	قيمة K	قيمة I
2	1 , 2	1
2	1 , 2	2
3	1 , 2 , 3	3
3	1 , 2 , 3	4

تمارين

١ ـــ اعـمـل على طباعة وتنفيذ البرامج الموجودة فى أشكال الفصل ، وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعديلات .

٢ ـــ بين الأخطاء إن وجدت فى كل من العبارات التالية :

أ ـــ
```
10    FOR      I = 1 TO 10
```
ب ـــ
```
10    FOR      I = 5 TO 1
```
جـ ـــ
```
10    NEXT
```
د ـــ
```
10    FOR      R = R TO N
```
هـ ـــ
```
10    FOR      J = I TO N STEP 4
```

٣ ـــ اعمل على تعديل البرامج التالية لكى يتم تنفيذها بدون أخطاء :

أ ـــ
```
10    FOR      I = I TO 5
20    PRINT    USING A$ ; I , I * I
30    NEXT     I
40    END
```

ب ـــ
```
10    READ     A , B , C
20    FOR      I = B TO C STEP 1
30    PRINT    I , A , B , C
40    NEXT     I
50    DATA     20 , - 10 , 4 , 5 , 8 , 6
60    END
```

جـ ــ

```
10   READ    N
20   FOR     I = 1 TO N
30   READ    A (I)
40   PRINT   A (I)
50   NEXT    J
60   DATA    9 , 4 , 2 , - 4 , 6 , 8 , 7 , 3 , 9
```

٤ ــ ماهى نتائج المعالجة (المخرجات) التى سنحصل عليها بعد تنفيذ البرامج التالية :

أ ــ

```
10   FOR     I = 1 TO 4
20   LET     A = I * 3
30   PRINT   I , A , I + A
40   NEXT    I
50   END
```

ب ــ

```
10   LET     A = 10
20   LET     B = 2
30   FOR     I = 1 TO A STEP B
40   PRINT   I , A * B + I
50   NEXT    I
60   END
```

جـ ــ

```
10   LET     A = 2
20   FOR     I = 10 TO 3 STEP A - 4
30   PRINT   I ;
40   NEXT    I
50   END
```

٥ ــ اعمل على تعديل السؤال الخامس فى الفصل العاشر مستخدماً تعليمة NEXT / FOR للحصول على إجمالى الدخل لجميع البائعين من كلا الصنفين ، وطباعته فى نهاية التقرير .

٦ ــ اعمل على تطوير برنامج لقراءة رقم القطعة والكمية المخزونة والكمية المطلوبة فى ثلاث مصفوفات أحادية البعد ، ومن ثم إجراء العمليات التالية :

١ ــ حساب الرصيد الجديد وتخزينه فى مصفوفة ذات بعد واحد .

٢ ــ طباعة قائمة بأرقام القطع وأرصدتها من الكميات ذات الرصيد السلبى ، وأخرى بالقطع ذات الرصيد الإيجابى وذلك كالتالى :

رقم القطعة	الرصيد	رقم القطعة	الرصيد
1123	15	3211	− 100
2114	10	0011	− 125
3215	20	1023	− 64
.	.	.	.
.	.	.	.
.	.	.	.

٣ ــ أوجد أدنى رصيد وأعلى رصيد فى كل قائمة .

ملاحظة : يمكن تخزين الأرصدة السالبة والأرصدة الموجبة فى مصفوفتين منفصلتين قبل طباعتهما .

أمثلة من البيانات المدخلة :

1123 , 65 , 50

2114 , 70 , 60

3211 , 200 , 300

0011 , 175 , 350

3315 , 60 , 40



المصفوفات ذات البعدين ومكوناتها وتعليمات الدوارة المتقدمة

مقدمة عن المصفوفات ذات البعدين :

تعرف أحادية البيانات المخزنة فى جدول معين عن طريق ارتباطها بخاصيتين ، تحدد الأولى السطر والثانية العمود ، فالجدول التالى يقسم الدارسين الملتحقين بدورة معينة فى معهد الإدارة العامة موزعين حسب خاصيتين : المناطق الجغرافية التى قدموا منها ، وتقديراتهم الفصلية .

التقدير	المنطقة الجغرافية			المجموع
	الوسطى	الشرقية	الغربية	
ممتاز	٥	٣	٢	١٠
جيد جداً	٦	٤	١	١١
جيد	٤	٣	٢	٩
مقبول	٤	١	١	٦
راسب	٢	١	١	٤
المجموع	٢١	١٢	٧	٤٠

فتحدد أحادية البيانات فى الجدول السابق بتقاطع أحد الأسطر وأحد الأعمدة، فالرقم ٦ مثلاً ينتج عن تقاطع السطر الممثل للتقدير «جيد جداً» والعمود الممثل للمنطقة «الوسطى» ، فنقول إن هناك ٦ دارسين من المنطقة الوسطى ذوى تقدير جيد جداً ....وهكذا لبقية الأرقام .

كيفية الإشارة إلى البيانات : كما مر ذكره فى الفصل السابق ، فإن أرقاماً لابد أن تستعمل للإشارة إلى محتويات الجدول من البيانات . ففى المصفوفة ذات البعد الواحد ، تم استخدام رقم واحد فقط ، أما فى المصفوفة ذات البعدين فلابد من استخدام رقمين منفصلين ، الأول للسطر والآخر للعمود . و يشبه نظام أرقام الغرف فى فندق يحتوى على أكثر من طابق واحد نظام الأرقام الذى نحن بصدد شرحه ، فالرقم «٣١٤» مثلا يرمز إلى الطابق الثالث ، حجرة رقم ١٤ . والفرق بين أرقام الحجرات وأرقام المصفوفة المعنية ، أن رقمى الأخيرة يفرق بينهما بالفاصلة « ، » فالرقم (١ ، ٤) يرمز إلى البيانات المخزنة فى الصندوق الناشىء عن تقاطع السطر الأول والعمود الرابع ، و يكون كل عدد الصناديق القابلة لتخزين البيانات يساوى عدد الأعمدة مضروباً × عدد الأسطر .

ونفرق هنا بين أرقام الصناديق ومحتوياتها من البيانات ، تماماً كما فى المصفوفة الواحدة . فالصندوق ذو الرقم (٢ ، ٢) فى المثال السابق يحتوى على ٤ دارسين من المنطقة الشرقية ذوى تقدير جيد جداً .

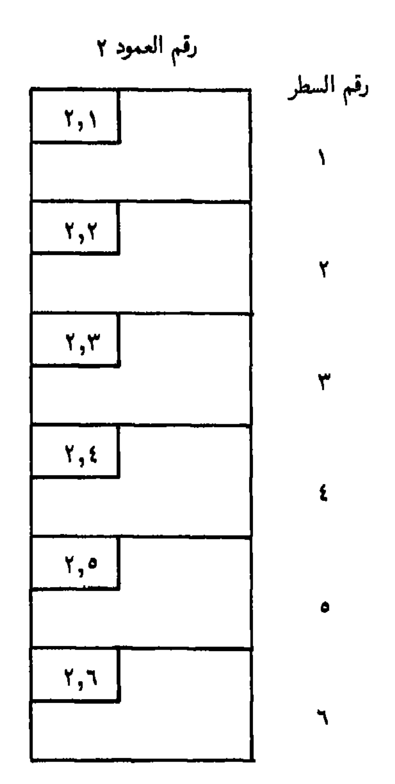
و يبين الشكل التالى أرقام الصناديق بالإضافة إلى محتوياتها من البيانات كما فى المثال السابق .

الأعمدة ←

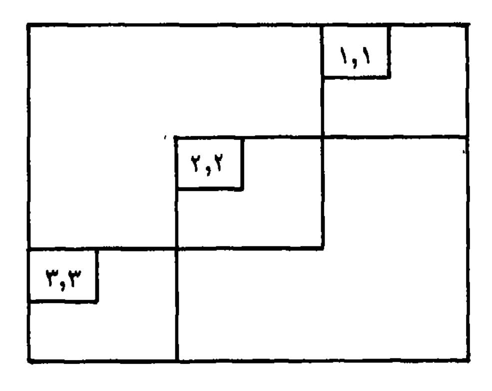
الأسطر ↓

التقدير	المنطقة ١	٢	٣	٤
١	١,١ ٥	٢,١ ٣	٣,١ ٢	٤,١ ١٠
٢	١,٢ ٦	٢,٢ ٤	٣,٢ ١	٤,٢ ١١
٣	١,٣ ٤	٢,٣ ٣	٣,٣ ٢	٤,٣ ٩
٤	١,٤ ٤	٢,٤ ١	٣,٤ ١	٤,٤ ٦
٥	١,٥ ٢	٢,٥ ١	٣,٥ ١	٤,٥ ٤
٦	١,٦ ٢١	٢,٦ ١٢	٣,٦ ٧	٤,٦ ٤٠

ففى المصفوفة السابقة ، ولنطلق عليها اسم «الدارس» ، أشرنا إلى رقم الصندوق التابع لتقاطع واحد من كل من الأسطر والأعمدة برقمين بينهما فاصلة « ، » . ولابد من تعريف مدلولات هذه الأرقام للحاسب فى البرنامج . فأرقام الأسطر من ١ إلى ٦ ترمز إلى التقديرات المختلفة والمجموع فى آخرها ، وأرقام الأعمدة من ١ إلى ٤ ترمز إلى المناطق والمجموع .



وكذلك أرقام الصناديق فى المصفوفة المرتبة قطرياً فى منتصفها باتجاه تسلسل الأرقام ، نجد أنها متماثلة فى رقم السطر ورقم العمود ، كما هو موضح فى الشكل التالى :



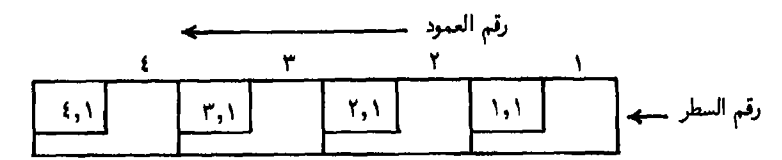
وذلك كله حسب الترتيب الظاهر فى الجدول . ففى مصفوفة «الدارس» يشير الصندوق رقم (٢,٣) إلى عدد الدراسين من المنطقة الشرقية ذوى التقدير «جيد» ، أو على العكس من ذلك ، عدد الدارسين الذين حصلوا على تقدير «جيد» وهم من المنطقة الوسطى . ونلاحظ هنا أن رقم السطر يأتى أولاً متبوعاً بفاصلة ثم رقم العمود .

ويمكننا أن نمثل رقم المنطقة أفقياً والتقدير عمودياً ، أى تدو ير الجدول بزاو ية ٩٠°. فيكون للجدول أر بعة أسطر وستة أعمدة . ولابد من تغيير أرقام الصناديق تبعاً لذلك .

العلاقة بين أرقام الأسطر والأعمدة :

إن معرفة كيفية تغيير أرقام الأعمدة والأسطر فى المصفوفة تؤدى إلى تسهيل برمجة المعالجات التى ستخضع لها البيانات المخزنة فيها .

فلو أخذنا سطراً أفقياً معيناً ، لوجدنا أن رقم السطر ثابت (الرقم الأول) فى حين يتغير رقم العمود (الرقم الثانى) كما هو موضح فى الشكل التالى :



ولو أخذنا عموداً معيناً ، لوجدنا أن أسلوب التغيير على العكس من ذلك ، فالرقم الأول ـــ رقم السطر ـــ يتغير فى حين يبقى الرقم الثانى ـــ رقم العمود ـــ ثابتاً ، كما هو موضح فى الشكل التالى :

رقم العمود ←

رقم السطر

١ ١,١

٢ ١,٢

٣ ١,٣

٤ ١,٤

وفى حالة تساوى عدد الأسطر مع عدد الأعمدة ، فإن هذا الخط القطرى من الصناديق يقسم المصفوفة إلى قسمين متساو يين فى عدد الصناديق .

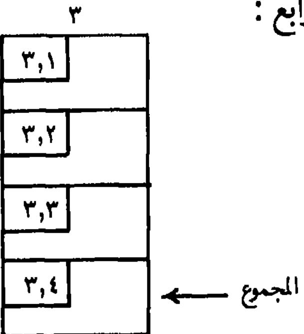
أما أرقام الصناديق فى القسمين المتساو يين ، فلتغيرها نظام خاص بها أيضا . ففى الشكل التالى :

٣,١ △	٢,١ *	١,١
٣,٢ ⊡	٢,٢	١,٢ *
٣,٣	٢,٣ ⊡	١,٣ △

نجد أن كل صندوقين متقابلين فى موقعيهما يحدث تبادل بين رقمى سطريهما وعموديهما ، كما هو مشار إليه بالرموز * ، △ ، ⊡

المجاميع الأفقية والعمودية :

قد يتطلب الأمر فى كثير من التطبيقات إيجاد مجاميع الأعمدة، والأسطر، وللحصول على هذه المجاميع، نتبع أيضاً نظاماً خاصاً. فللعمود التالى رقم ٣، يوضع المجموع فى السطر الرابع :



وتكون المعادلة لإيجاد مجموع القيم المخزنة فى الصناديق التابعة لهذا العمود :

(٣,٤) = (٣,١) + (٣,٢) + (٣,٣)

ونلاحظ تغير أرقام الأعمدة والأسطر على النحو التالى :

أ) عدم تغيير رقم العمود .

ب) زيادة سطر صندوق المجموع ب «١» عن آخر رقم للأسطر .

و يكون تغيير الأرقام على العكس من ذلك عند إيجاد المجموع لسطر معين ، كما فى السطر التالى رقم ٢

٤	٣	٢	١	
٤,٢	٣,٢	٢,٢	١,٢	٢

↑ المجموع

وهكذا يكون تغيير أرقام صناديق مجاميع الأعمدة والأسطر فى أية مصفوفة ، ومن الممكن مراجعة الشكل ١٢ ــ ١ فى بداية هذا الفصل للتدقيق فى كيفية تغيير أرقام صناديق التخزين فى كل سطر وعمود .

مقدمة عن الدوارة المركبة وكيفية استخدامها للتحكم فى عمليات المصفوفات ذات البعدين :

لاحظنا فى الفصل السابق كيف سهلت تعليمات الدوارة البسيطة معالجة البيانات المخزنة فى المصفوفة ذات البعد الواحد . وتم ذلك عن طريق استخدام عداد الدوارة للدلالة على رقم الصندوق التابع لمصفوفة معينة ، حيث يوضع هذا الرقم بين قوسين عقب اسم المصفوفة ، كالتالى N (I) حيث يرمز I إلى رقم الصندوق الذى يتغير من ١ إلى عدد الصناديق الكلى ، وهو نفسه عداد الدوارة .

أما فى حالة المصفوفة ذات البعدين، فنحن بحاجة هنا إلى رقمين للدلالة على صندوق معين، الأول للدلالة على رقم السطر، والثانى للدلالة على رقم العمود، وتتم الإشارة إلى صندوق معين فى الدوارة ذات البعدين عن طريق وضع الرقمين ــ العمود والسطر ــ بين قوسين، ومفصولين بالفاصلة «،»، كالتالى :

S (1,2) للدلالة على الصندوق فى السطر الأول والعمود الثانى فى المصفوفة S.

وكذلك يمكن استخدام متغيرات وإسناد قيم لها كالتالى :

S (I,J) حيث يتغير I من ١ إلى عدد الأسطر و J من ١ إلى عدد الأعمدة.

وكما كان الشأن مع الدوارة البسيطة، فإن الدوارة المركبة يمكن استخدامها فى حالات تستدعى معالجة بيانات دون تخزينها فى مصفوفة ذات بعدين، وإن كان هذا الأخير يمثل أشهر وأهم تطبيق لها.

وقد مر بنا فى الفصل السابق كيف يمكن أن تتوالى الدورات البسيطة فى برنامج واحد، وذلك على الشكل التالى :

```
┌  100 FOR I 1 TO 10
└  150 NEXT I

┌  160 FOR J = 1 TO 20 STEP 2
└  200 NEXT J
          ⋮
```

وهكذا

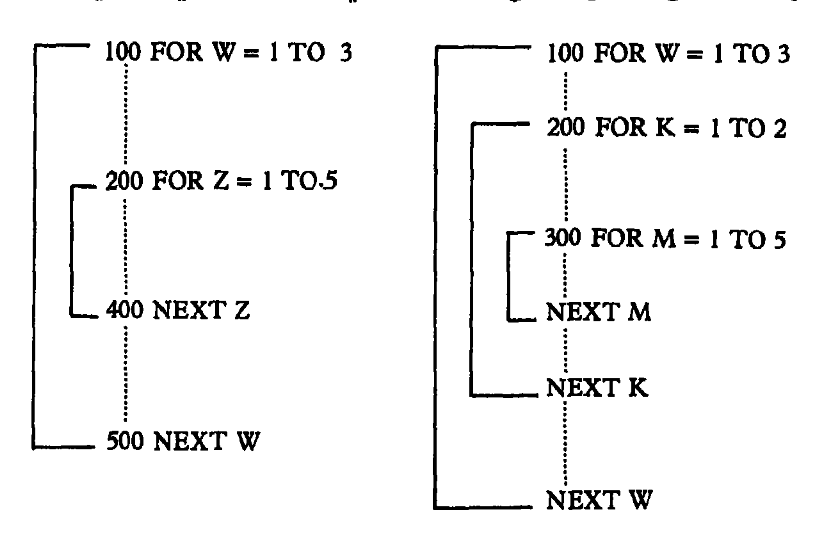
فماذا يحدث لو وضعنا دوارة بسيطة داخل أخرى؟ وكيف تكون العلاقة بين عدادى الدوارتين؟

لاحظ أن الشكل التالى للتداخل بين الدوارات غير مقبول حسب قواعد البرمجة :

```
100  FOR  J = 1  TO  3
120  FOR  K = 1  TO  5  STEP  2
150  NEXT  J

250  NEXT  K
```

لذلك فإن الشكل الصحيح للدوارة المركبة أن تحتوى كل واحدة منها الأخرى احتواء تاما ، ومن الممكن تداخل أكثر من دوارتين ، كما فى المثالين التاليين :



وترمز النقاط الظاهرة بين العبارات إلى إمكانية وجود عبارات لغة البيسك أو عدمه ، أى أن عبارات LET ، READ ، PRINT....إلخ قد تتناثر، حسب طبيعة البرنامج ، بين عبارات الدوارة المركبة . مع الحرص على عدم التشعب من داخل إحدى الدوارات إلى آخرها ، وإنما يكون التشعب إلى بداية الدوارة أو إلى نهايتها حسب منطق البرنامج .

تغير قيم عدادات الدوارة : وحتى نفهم كيفية استخدام الدوارة المركبة لابد من معرفة العلاقة بين القيم التى تأخذها الدوارة، ولنأخذ المثال التالى :

```
100   FOR   I = 1  TO  3
 :
150   FOR   J = 1  TO  2
 :
200   NEXT   J
 :
250   NEXT   I
```

إن قيم I للدورة الكاملة هى كالتالى : ١، ٢، ٣

وقيم J للدورة الكاملة هى كالتالى : ١، ٢

و يتم تتابع قيمة I و J كالتالى :

لكل قيمة واحدة من قيم I، تتم J دورتها الكاملة ابتداء من أول قيمة لها «١» وانتهاء بآخر قيمة لها «٢» ، و يوضح ذلك الشكل التالى :

قيمة I	قيمة J	
١	١	الدورة الأولى
١	٢	
٢	١	الدورة الثانية
٢	٢	
٣	١	الدورة الثالثة
٣	٢	

لاحـظ :

(١) أن عدد الدورات الكاملة للدوارة المركبة تتحكم فيها قيم عداد الدوارة الخارجية.

(٢) أن عـدد المرات التى يتم فيها تنفيذ العبارات الظاهرة بين عبارتى FOR و NEXT فى الـدوارة الداخلية هى ناتج ضرب القيمتين النهائيتين لعدادى الدوارة الداخلية والدوارة الخارجية = ٣×٢= ٦ فى هذه الحالة.

(٣) أن عـدد المـرات التـى يتـم فيها تنفيذ العبارات الواردة بين FOR الأولى و FOR الثانية يساوى ٣ــ القيمة النهائية لعداد الدوارة الخارجية.

(٤) أن قيمة عداد الدوارة الداخلية «J» تساوى ٣ عند انتقال التنفيذ من عبارة NEXT J إلى عبارة NEXT I.

(٥) بـعـد الانـتـهـاء مـن تـنـفيذ NEXT I للمرة الأخيرة، تكون I = ٤ و J = ٣ وذلك بـزيـادة «١» على الـقـيمة النهائية المعلنة فى عبارتى FOR، أو بزيادة الرقم الذى يظهر بعد STEP فى حالة اختلافه عن «١».

إذا أضفنا تعليمات الطباعة PRINT لمعرفة قيمة I، J أثناء تنفيذ الدوارة كالتالى :

```
100 FOR I = 1 TO 3
110 PRINT « 110 رقم السطر I = » ; I ; « J = » = J
150 FOR J = 1 TO 2
160 PRINT « 160 رقم السطر » ; « I = » , I ; « J = » ; I
200 NEXT J
210 PRINT « 210 رقم السطر » ; « I = » = I ; « J = » ; J
250 NEXT I
260 PRINT « 260 رقم السطر » ; « I = » ; I ; « J = » ; J
```

ولكان الناتج بعد تنفيذ البرنامج السابق :

ملاحظات	الناتج			
بداية الدوارة المركبة ـ الدوارة الخارجية	J = 0	I = 1	رقم السطر	110
بداية الدوارة الداخلية للمرة الأولى	J = 1	I = 1	رقم السطر	160
	J = 2	I = 1	رقم السطر	160
انتهاء دورة كاملة للدوارة الداخلية	J = 3	I = 1	رقم السطر	210
بداية الدورة الثانية للدوارة الخارجية	J = 3	I = 2	رقم السطر	110
بداية الدوارة الداخلية للمرة الثانية	J = 1	I = 2	رقم السطر	160
	J = 2	I = 2	رقم السطر	160
انتهاء الدورة الكاملة للدوارة الداخلية	J = 3	I = 2	رقم السطر	210
بداية الدورة الثالثة للدوارة الخارجية	J = 3	I = 3	رقم السطر	110
بداية الدوارة الداخلية للمرة الثالثة	J = 1	I = 3	رقم السطر	160
	J = 2	I = 3	رقم السطر	160
انتهاء الدورة الثالثة للدوارة الخارجية	J = 3	I = 3	رقم السطر	210
انتهاء الدوارة المركبة	J = 3	I = 4	رقم السطر	260

علاقة أرقام عدادات الدوارة بالمصفوفة ذات البعدين : لو دققنا النظر إلى السطر ١٦٠، الواقع داخل الدوارة الداخلية، وتتبعنا تغير قيم I و J وقارناه بأرقام الصناديق فى المصفوفة لوجدناه كالتالى :

رقم السطر ← رقم العمود ←

	I	J	رقم السطر
السطر الأول	1	1	1
	1	2	
السطر الثانى	2	1	2
	2	2	
السطر الثالث	3	1	3
	3	2	

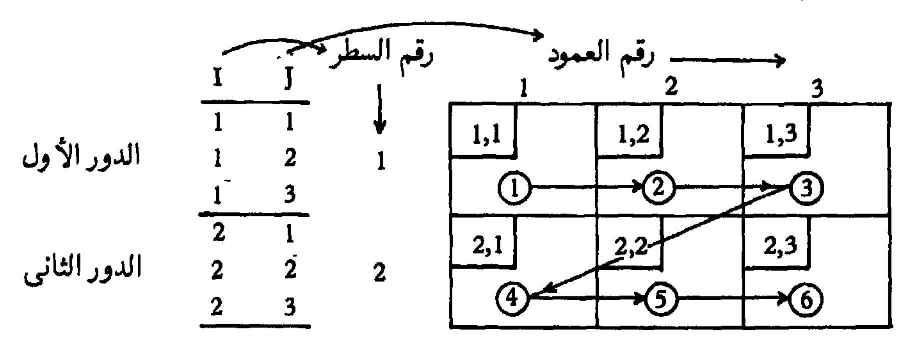
1	2
1,1 ①	1,2 ②
2,1 ③	2,2 ④
3,1 ⑤	3,2 ⑥

أى أن الدوارة المركبة بوضعها الحالى يمكن أن تستخدم لتمثيل مصفوفة ذات ثلاثة أسطر وعمودين، و يتم ذلك عن طريق وضع الفاصلة «,» بين العدادين I , J وإضافة اسم المصفوفة، «S» مثلا، فتكون :

S (I , J). ولاحظ تسلسل الإشارة إلى الصناديق :

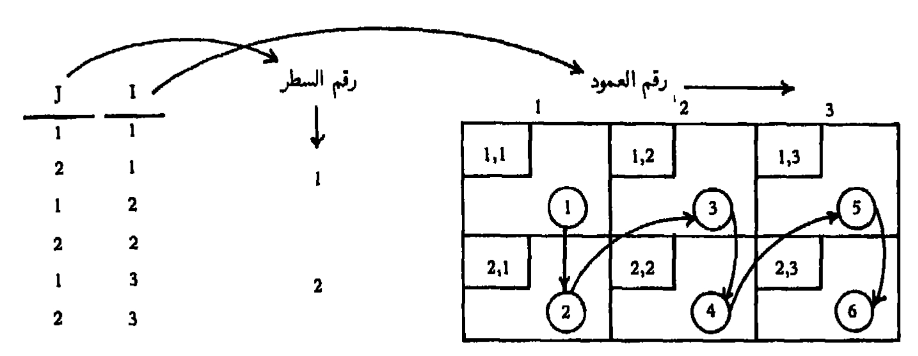
① ← ② ← ③ ← ④ ← ⑤ ← ⑥ ، وذلك وفق تسلسل القيم المزدوجة I و J.

لاحظ أن رقم السطر فى المصفوفة يمثله عداد الدوارة الخارجية ورقم العمود يمثله عداد الدوارة الداخلية، و بناء على ذلك، فإن التعامل مع مصفوفة ذات سطرين وثلاثة أعمدة، أى على العكس من المصفوفة السابقة، يتطلب جعل القيمة النهائية لعداد الدوارة الخارجية تساوى ٢ وقيمة عداد الدوارة الداخلية تساوى ٣. وفى هذه الحالة تكون الإشارة إلى الصناديق كالتالى :



وهنـاك حـل آخر وهو إبقاء قيمتى العدادين كما هما وإبدال مكان I مكان J بين القوسين، بحيث تشير I إلى رقم العمود و J إلى رقم السطر.

S(J , I)، كما هو مبين فى الشكل التالى :



وفى كـل الأحـوال، فـإن المحافظة على تسلسل الإشارة إلى الصناديق بشكل أفقى تؤدى إلى وضوح أكثر وإلى أسلوب أيسر، لتخزين ومعالجة البيانات الخاصة بالمصفوفات ذات البعدين.

فـمـلاحـظة هذا التسلسل إذن أمر مهم جداً عند البدء فى عمليات تخزين البيانات فى الصناديق ومعالجتها.

وكـمـا كـان الـشأن مع المصفوفة ذات البعد الواحد، فإن تعليمات بيسك قد تظهر قبل اسم المصفوفة، وذلك كما يلى :

```
READ     S  (I,J)
INPUT    S  (I,J)
PRINT    S  (I,J)
LET      T = T + S (I,J)
```

وسـنـتـطـرق الآن إلى كـيـفية تطوير برنامج باستخدام تعليمات الدوارة المركبة فى العمليات المصاحبة لاستخدام المصفوفات ذات البعدين.

تطوير برنامج باستخدام تعليمات الدوارة المركبة مع المصفوفات ذات البعدين :

سنتعرض الآن لبرنامج مبسط لتوضيح كيفية التعامل مع المصفوفة ذات البعدين

مثال : (١٢ ـــ ١) :

تؤجر المدينة الرياضية ثلاث صالات ، منها واحدة رئيسية واثنتان ذواتا حجم أصغر ، فرعيتان . وهناك ناديان فى المدينة : نادى الشباب ، ونادى الوحدة .

الهدف : معرفة مجموع الساعات لكل نادٍ ولكل صالة ، والمجموع الكلى لعدد الساعات ، وذلك كل شهر .

أولاً ـــ خطوات الحل :

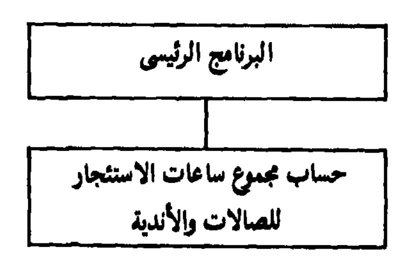
١ ـــ الحصول على ساعات الإيجار لكل نادٍ فى كل صالة ، وتخزينها فى مصفوفة : عدد الأسطر ٢ يساوى عدد النوادى الرياضية ، وعدد الأعمدة ٣ يساوى عدد الصالات .

٢ ـــ حساب مجموع الأعمدة والأسطر .

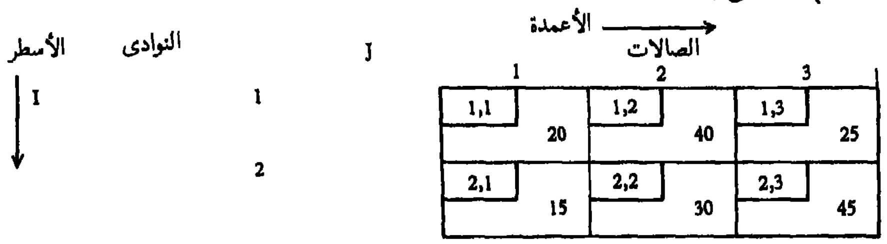
٣ ـــ حساب المجموع الكلى لعدد الساعات .

٤ ـــ طباعة النتائج .

ثانياً ـــ الهيكل الهرمى :



وتوضح هذه البيانات فى مصفوفة مبينة فيها أرقام الصناديق (و يطلق عليها خلايا Cells) كالتالى :



ويمكن قراءة هذه المصفوفة وتخزينها تحت اسم HR كما فى الجزء التالى :

```
FOR      I = 1 TO 2
FOR      J = 1 TO 3
READ     HR (I,J)
NEXT     J
NEXT     I
DATA     20, 40, 25
DATA     15, 30, 45
```

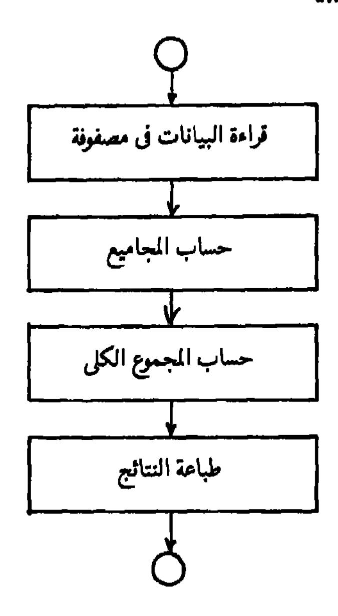
لاحظ التسلسل الأفقى لظهور البيانات فى عبارتى DATA ، وذلك وفق استخدام عدادى الدوارة المركبة . ولو غيرنا سطر القراءة ليصبح كالتالى :

```
READ HR (J,I)
```

لتمت قراءة البيانات فى مصفوفة ذات ثلاثة أسطر وعمودين كما هو موضح فيمايلى :

1,1 20	1,2 15
2,1 40	2,2 30
3,1 25	3,2 45

ثالثا ــ نمط البرمجة التركيبية :



رابعاً ــ البرمجة :

ونورد هنا الشرح التفصيلى لتنفيذ هذه الأجزاء وما يصاحبها من عبارات البرمجة :

١ ــ **الحصول على البيانات** : كانت ساعات الاستئجار من قبل الأندية للصالات الثلاث لشهر رجب ١٤٠٧ هـ كما يلى :

النادى	الصالات الرئيسية	الصالات الفرعية ١	الفرعية ٢
نادى الشباب	20	40	25
نادى الوحدة	15	30	45

أما إذا تم تغيير السطرين الخاصين بـ For ليصبحا كالتالى :

```
FOR I = 1 TO 3
FOR J = 1 TO 2
```

فإن ترتيب البيانات فى عبارات DATA سيبقى كما هو فى الوضع السابق، إلا أن المصفوفة سيكون لها ثلاثة أسطر ممثلة للصالات، وعمودان ممثلان للأندية، أى أن المصفوفة رقمت بعكس الوضع السابق، كما هو موضح فى الشكل التالى :

الصالات I — الأندية J

I	1	2
1	1,1 — 20	1,2 — 15
2	2,1 — 40	2,2 — 30
3	3,1 — 25	3,2 — 45

وكما تم ذكره سابقاً فى هذا الفصل فإن الوضع الأول هو الأوضح والأسهل لتخزين البيانات فى المصفوفات ومعالجتها. وهو ماسيتم اتباعه فى هذا الفصل.

٢ ــ الحصول على المجاميع : هناك أكثر من طريقة للحصول على المجاميع ومن هذه الطرق :

أ) تخزين المجاميع فى متغيرات مستقلة عن المصفوفة.

ب) تخزين المجاميع فى المصفوفة نفسها أى فى السطر الثالث والعمود الرابع.

وسنبحث الأسلوب الأول هنا، وسنتعرض للأسلوب الثانى بعد تطوير البرنامج لهذا المثال بشكل كامل.

فلو فرضنا أن المتغيرات C3, C2, C1, R2, R1 ستحتوى على مجاميع عدد الساعات للأندية والصالات على التوالى، فستكون الحسابات كالتالى :

للأسطر الأندية

R1 = HR (1,1) + HR (1,2) + HR (1,3)

R2 = HR (2,1) + HR (2,2) + HR (2,3)

للأعمدة الصالات

C1 = HR (1,1) + HR (2,1)

C2 = HR (1,2) + HR (2,2)

C3 = HR (1,3) + HR (2,3)

ولـكن هذا الأسلوب سيبدو عديم الجدوى إذا ما افترضنا التعامل مع مصفوفة ذات بـعـدين كـبـيـريـن، ١٠٠ سـطر و ٥٠ عموداً مثلاً . ومن الممكن اتباع ماتم شرحه عن العلاقة بين أرقام الصناديق، فتصبح المعادلات مع عبارات البرمجة كالتالى :

```
10 REM حساب مجموع الا سطر
20 FOR J =1 TO 3
30 R1 = R1 + HR(1,J)
40 R2 = R2 + HR(2,J)
50 NEXT J
60 REM حساب مجموع الا عمده
70 FOR I =1 TO 3
80 C1 = C1 + HR(I,1)
90 C2 = C2 + HR(I,2)
100 C3 = C3 + HR(I,3)
110 NEXT I
```

لاحـظـنـا ثـبات أرقام الأسطر وتغير أرقام الأعمدة فى الجزء الأول الخاص بحساب مجاميع الأسطر، وعلى العكس من ذلك فى الجزء الخاص بحساب مجاميع الأعمدة. ومن الممكن تتبع تنفيذ هذا الجزء كالتالى، ونبدأ بحساب مجاميع الأسطر :

الدورة	I	J	قيمة HR(I,J)	الأسطر R1	R2
1	1	1	20	20	
	2	1	15		15
2	1	2	40	60	
	2	2	30		45
3	1	3	25	85	
	2	3	45		90
المجموع النهائى				85	90

حيث يتم فى كل دورة إضافة قيمتين من قيم المصفوفة HR إلى متغيرات المجاميع ذات العلاقة بكل سطر .

أما مجاميع الأعمدة فيمكن تتبعها كالتالى :

الدورة	I	J	HR (I,J)	الأعمدة C1	C2	C3
1	1	1	20	20		
	1	2	40		40	
	1	3	25			25
2	2	1	15	35		
	2	2	30		70	
	3	3	45			70
المجموع				35	70	70

حيث يتم إضافة ثلاث قيم من المصفوفة فى كل دورة إلى قيمة المجاميع ذات العلاقة بكل عمود.

لاحظ اتجاه الأسهم بين قيم HR (I,J) والقيم المتجمعة فى الأسطر والأعمدة والإشارة الحسابية من + أو =. ففى الشكل السابق خزنت القيمة 20 فى C1 فى الدورة الأولى، ثم أضيف لها 15 فى الدورة الثانية، وهكذا مع بقية قيم المتغيرات الخاصة بمجاميع الأسطر والأعمدة.

٣ ــ حساب المجموع الكلى : يتم تخزين كل قيمة من قيم الصناديق فى متغير جامع، كما هو موضح فى الجزء التالى من البرنامج :

```
FOR     I = 1 TO 2
FOR     J = 1 TO 3
T   =   T  +  HR (I,J)
NEXT    J
NEXT    I
```

٤ ــ كتابة النتائج : إن طباعة النتائج تحتاج إلى قليل من العناية. وسنكتفى هنا بطباعة أرقام الساعات كما فى الجدول، يليها أرقام المجاميع. وسنقدم فيما بعد برنامجاً لطباعة العناوين الموضحة لأرقام المصفوفة.

ويمكن تجميع الأجزاء السابقة فى برنامج كامل كما فى الشكل التالى :

شكل (١٢ — ١)

برنامج لحساب عدد الساعات المستأجرة من قبل الأندية لصالات المدينة الرياضية

```
10 REM البرنامج الرئيسي
20 GOSUB 90
30 END
90 REM البرنامج الفرعي
100 REM      HR قراءة البيانات في مصفوفة
110 FOR I = 1 TO 2
120 FOR J = 1 TO 3
130 READ HR(I,J)
140 NEXT J
150 NEXT I
160 DATA 20,40,25
170 DATA 15,30,45
180 REM حساب مجاميع الاسطر
190 FOR J = 1 TO 2
200 R1= R1 + HR(1,J):REM مجموع السطر الاول
210 R2= R2 + HR(2,J):REM مجموع السطر الثاني
220 NEXT J
230 REM حساب مجاميع الاعمدة
235 FOR I = 1 TO 3
240 C1 = C1 + HR(I,1):REM مجموع العمود الاول
250 C2 = C2 + HR(I,2):REM مجموع العمود الثاني
260 C3 = C3 + HR(I,3):REM مجموع العمود الثالث
270 NEXT I
280 REM حساب المجموع الكلي للساعات
290 FOR I = 1 TO 2
300 FOR J = 1 TO 3
310 T = T + HR(I,J)
320 NEXT J
330 NEXT I
340 REM طبع النتائج والبيانات المدخلة
350 FOR I = 1 TO 2
360 FOR J = 1 TO 3
370 PRINT USING "##     ";HR(I,J);
380 NEXT J
390 PRINT " ":REM للانتقال للسطر الجديد
400 NEXT I
410 PRINT R1,"مجموع ساعات نادي الشباب"
420 PRINT R2,"مجموع ساعات نادي الوحدة"
430 PRINT C1,"مجموع ساعات الصالة الكبرى"
440 PRINT C2,"مجموع ساعات الصالة الفرعية الاولى"
450 PRINT C3,"مجموع ساعات الصالة الفرعية الثانية"
460 PRINT T,"المجموع الكلي للساعات المستاجرة"
470 RETURN
```

وفى حالة تنفيذ البرنامج سنحصل على النتائج التالية ·

```
20      40      25
15      30      45
 60        مجموع ساعات نادي الشباب
 45        مجموع ساعات نادي الوحدة
 35        مجموع ساعات الصالة الكبرى
 70        مجموع ساعات الصالة الفرعية الاولى
 70        مجموع ساعات الصالة الفرعية الثانية
 175       المجموع الكلي للساعات المستاجرة
```

ملاحظات على البرنامج :

١ ــ إن الأجزاء الثلاثة الأولى من البرنامج الفرعى يمكن أن تجمع حقاً فى دوارة مركبة واحدة. وتوضيح ذلك أنه عند قراءة كل قيمة من قيم ساعات الاستئجار يتم تخزينها فى الخلية المناسبة، وإضافتها إلى مجموع السطر والعمود ذَوَى العلاقة وإلى القيمة الكلية للساعات.

ويمكن تعديل البرنامج السابق ليتوافق مع الأسلوب الجديد بإزالة الأسطر التالية :

300, 290, 270, 235, 220, 190, 150, 140

٢ ــ إذا أردنا تخزين مجموع الأسطر فى العمود الأخير ومجموع الأعمدة فى السطر الأخير، يصبح شكل المصفوفة النهائية بعد إجراء عمليات الجمع كالتالى :

النادى	الصالات الرئيسية	الفرعية ١	الفرعية ٢	المجموع
الشباب	1,1 — 20	1,2 — 40	1,3 — 25	1,4 — 85
الوحدة	2,1 — 15	2,2 — 30	2,3 — 45	2,4 — 90
المجموع	3,1 — 35	3,2 — 70	3,3 — 70	3,4 — 175

وباستخدام العلاقة بين تغيرات أرقام الصناديق، يمكن تطوير العبارات الخاصة بتجميع الأسطر والأعمدة كالتالى :

```
REM        حساب مجموع الأسطر والأعمدة
HR (I,4) = HR (I,4) + HR (I,J)
HR (3,J) = HR (3,J) + HR (I,J)
REM        حساب المجموع الكلى لساعات الاستئجار
HR (3,4) = HR (3,4) + HR (I,J)
```

ولإحلال هذه الأسطر فى البرنامج مكان الأسطر الأخرى، احذف الأسطر من 200 - 260، ومن ثم أضف الأسطر الجديدة.

٣ ــ إذا أريد طباعة الجداول كما فى شكلها السابق، يجب تخزين أسماء الأندية مع كلمة «المجموع» فى مصفوفة مستقلة ذات بعد واحد، ذلك لأن قوانين لغة البيسك تمنع الجمع بين بيانات حسابية وغير حسابية (حرفية) فى نفس المصفوفة. ويمكن تحقيق ذلك كما فى الجزء التالى :

```
REM        قراءة أسماء الأندية كلمة المجموع
FOR K = 1 TO 3
READ N$ (K)
NEXT K
DATA «المجموع» و «الوحدة» و «الشباب»
```

وهذا التخزين لأسماء الأندية ضرورى، لنتمكن من طباعة اسم النادى قبل طباعة الساعات الخاصة به . أما طباعة أسماء الصالات الرياضية فأمر سهل ، إذ يمكن طباعتها فى سطر واحد قبل بدء طبع أسماء الأندية والصالات.

ولطباعة الجدول، من الأفضل أن تحدد أماكن طبع العناوين والأرقام على الشاشة، كما فى الشكل التالى :

النادى	الصالات			
	الرئيسية	الفرعية ١	الفرعية ٢	المجموع
الشباب	xx	xx	xx	xxx
الوحدة	xx	xx	xx	xxx
المجموع	xx	xx	xx	xxx

وسنبدأ بكتابة العناوين ثم طباعة الجدول .

التعليمات التالية تطبع السطر الأول والثانى من العناوين :

```
PRINT TAB (33); «الصالات»
PRINT TAB (10); « النادى» TAB (22) ; «الرئيسية»; TAB (32);
«الفرعية ١»; TAB( 42 ) ; «الفرعية ٢»; TAB (52); «المجموع»;
```

والتعليمات التالية تقوم بطباعة الجدول مع أسماء النوادى :

```
FOR I = 1 TO 3
PRINT TAB (12); N$ (I) ;
FOR J = 1 TO 4
PRINT   «    »    ; HR (I,J);
NEXT J
PRINT
NEXT I
```

ولتعديل البرنامج فى الشكل (١٢ـ١)، احذف الأسطر من 340 إلى 460 وأضف الأسطر البديلة السابقة، لاحظ أن البرنامج فى هذا الشكل قد عدل أكثر من مرة إلى الآن، وسنورده هنا فى شكله النهائى :

شكل (١٢ ــ ١أ)

برنامج لقراءة البيانات فى مصفوفة ذات بعدين وحساب
مجاميع الأسطر والأعمدة وطباعة المصفوفة مع النتائج

```
10 REM البرنامج الرئيسي
20 GOSUB 40
30 END
40 REM البرنامج الفرعي
50 REM     HR قراءة البيانات في مصفوفة
60 FOR I = 1 TO 2
70 FOR J = 1 TO 3
80 READ HR(I,J)
90 NEXT J
100 NEXT I
110 DATA 20,40,25
120 DATA 15,30,45
130 REM قراءة اسماء الاندية وكلمة المجموع
140 FOR K = 1 TO 3
150 READ N$(K)
160 NEXT K
170 DATA " الشباب","الوحدة ","المجموع"
180 FOR I= 1  TO 2
190 FOR J = 1 TO 3
200 REM  حساب مجاميع الاسطر والاعمدة
210 HR(I,4) = HR(I,4) + HR(I,J)
220 HR(3,J) = HR(3,J) + HR(I,J)
230 REM حساب المجموع الكلي للساعات المستاجرة
240 HR(3,4) = HR(3,4) + HR(I,J)
250 NEXT J
260 NEXT I
270 REM طباعة السطر الاول والثاني من العناوين
280 PRINT TAB(33);"الصالات"
290 PRINT TAB(12);"النادي";TAB(25);"الرئيسية";TAB(37);"الفرعية ١";TAB(51);
300 PRINT "الفرعية ٢";TAB(65);"المجموع"
310 REM طباعة الجدول مع اسماء النوادي
320 FOR I = 1 TO 3
330 PRINT TAB(12);N$(I);
340 FOR J = 1 TO 4
350 PRINT "          ";HR(I,J);
360 NEXT J
370 PRINT " "
380 NEXT I
390 RETURN
```

وعند تنفيذ البرنامج سنحصل على النتائج التالية :

النادي	الصالات			
	الرئيسية	الفرعية ١	الفرعية ٢	المجموع
الشباب	20	40	25	85
الوحدة	15	30	45	90
المجموع	35	70	70	175

وسنضيف الآن هدفاً آخر للمثال السابق يدعو إلى حساب ما سيدفعه كل نادٍ مقابل ساعات الاستئجار، إذا علم أن كل نادٍ سيدفع مبلغاً محدداً لكل ساعة، كما هو مبين فى الجدول التالى :

إجمالى ساعات الاستئجار فى الصالات الرياضية بالريال السعودى .

النادى	الصالات		
	الرئيسية	الفرعية ١	الفرعية ٢
الشباب	400	230	190
الوحدة	320	220	200

حيث يقرر هذا المبلغ مركز النادى فى الدورى الممتاز، وقد فاز نادى الشباب بالدورى لهذه السنة، وبناء على ذلك فأجر الساعة له أعلى من نادى الوحدة الذى حقق المركز الرابع، إلا فى الصالة الفرعية ٢ .

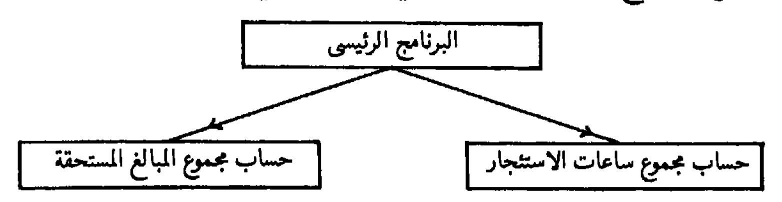
ولتحقيق هذا الهدف، نضيف الخطوات التالية :

٥ ــ قراءة مصفوفة الأجر للساعة الواحدة من ساعات الاستئجار.

٦ ــ حساب مجموع ماسيدفعه كل نادٍ والدخل لكل صالة .

٧ ــ الطباعة .

وسنجعل للبرنامج السابق جزأين رئيسيين كما هومبين فى الهيكل الهرمى التالى :



وقد تم تنفيذ الجزء الأول وسنتم الآن الجزء الثانى وفق الخطوات السابقة وباستخدام نفس نمط البرمجة التركيبية للجزء الأول.

٥ ــ قراءة المصفوفة :

```
800 FOR I = 1 TO 2
810 FOR J = 1 TO 3
820 READ RT (I,J)
830 NEXT J
840 NEXT J
850 DATA 400, 230, 190
860 DATA 320, 220, 200
```

٦ ــ حساب المجموع المستحق :

```
895 REM
900 FOR I = 1 TO 2
910 FOR J = 1 TO 3
920 PY (I,J) = HR (I,J) * RT (I,J)
930 PY (I,4) = PY (I,4) + PY (I,J)
940 PY (3,J) = PY (3,J) + PY (I,J)
950 TT = TT + PY (I,J)
960 NEXT J
970 NEXT I
```

٧ ــ الطباعة :

```
1000 REM
1010 PRINT TAB ( 40) ;  «مجموع مستحقات النوادى والإيرادات»
1020 PRINT TAB (10);          «النوادى»
TAB (52);
1050 FOR I = 1 TO 3
1060 PRINT TAB (12); N$ (I);
1070 FOR J = 1 TO 4
1080 PRINT «   »; PY (I,J) ;
1090 NEXT J
1100 PRINT
1110 NEXT I
1200 RETURN
```

ولتنفيذ هذا الجزء ، أضف الأسطر السابقة إلى البرنامج فى الشكل (١٢ ــ ١أ)، مع العبارة التالية :

```
25        GOSUB        800
```

وفى حالة تنفيذ البرنامج المعدل سنحصل على النتائج التالية :

الصالات

النادي	الرئيسية	الفرعية ١	الفرعية ٢	المجموع
الشباب	20	40	25	85
الوحدة	15	30	45	90
المجموع	35	70	70	175

مجموع مستحقات النوادي وايرادات الصالات

الصالات

النوادي	الرئيسية	الفرعية ١	الفرعية ٢	المجموع
الشباب	20	40	25	85
الوحدة	15	30	45	90
المجموع	35	70	70	175

إجراء العمليات الرياضية على المصفوفات
واستخدامات تعليمة مصفوفة MAT :

تعرضنا فى الجزء السابق من هذا الفصل لكيفية معالجة البيانات المخزنة فى مصفوفة ذات بعدين عن طريق الدوارة المركبة، لكن التطبيقات العلمية والرياضية قد تتطلب معالجة معقدة على المصفوفات، بحيث تجعل الاعتماد على تعليمات الدوارة المركبة قليل الجدوى.

وسنتعرض فى هذا الجزء من الفصل لاستخدام تعليمة MAT ، والتى تسهل المعالجة الخاصة بالمصفوفات. وتقسم هذه الاستعمالات إلى ثلاثة أقسام رئيسية :

١ ـ استخدامات فى الإدخال والإخراج.

٢ ـ استخدامات فى العمليات الحسابية من إضافة أو ضرب مصفوفة فى رقم واحد محدد، أو جمع وضرب وطرح مصفوفتين.

٣ ـ استخدامات فى عمليات خاصة مثل إيجاد منقول ومعكوس المصفوفة.

وسيتم شرح جميع هذه الاستخدامات مع التركيز على الاستخدام الأول والثانى لمناسبتهما لموضوع الكتاب.

تطوير برنامج باستخدام تعليمة MAT :

مثال (١٢ ـ ٢) :

الهدف : دائرة الأرصاد الجوية فى مدينة معينة ترغب فى إيجاد متوسط درجات الحرارة لكل أسبوع من الأسابيع الأربعة ومن ثم المتوسط العام للأسابيع الأربعة.

أولاً ـ خطوات الحل :

١ ـ الحصول على درجات الحرارة للأسبوع.

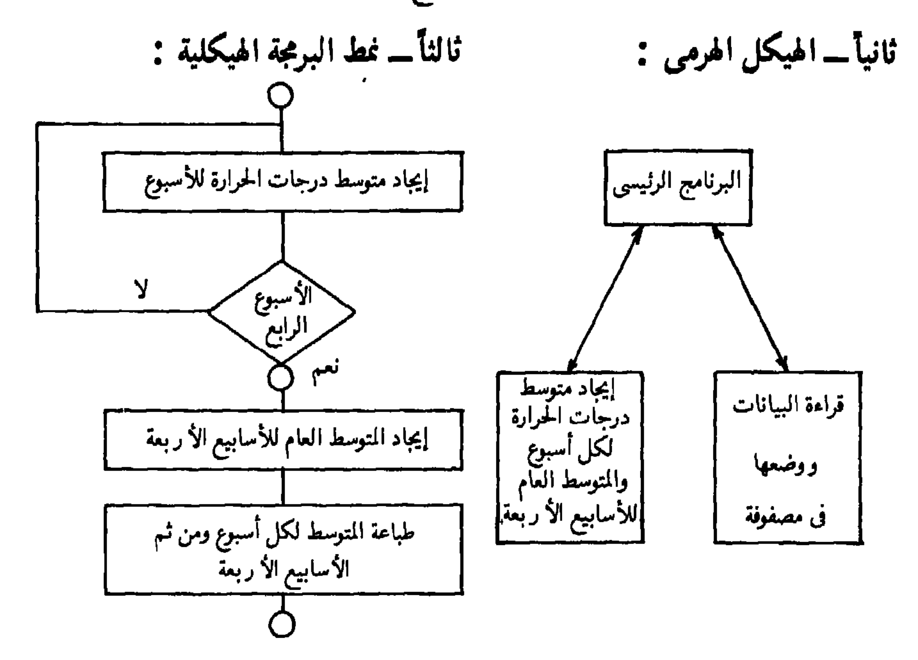
٢ ـ إيجاد متوسط درجات الحرارة للأسبوع.

٣ ـ تكرار الخطوتين ١ و ٢ للأسابيع الأ ربعة.

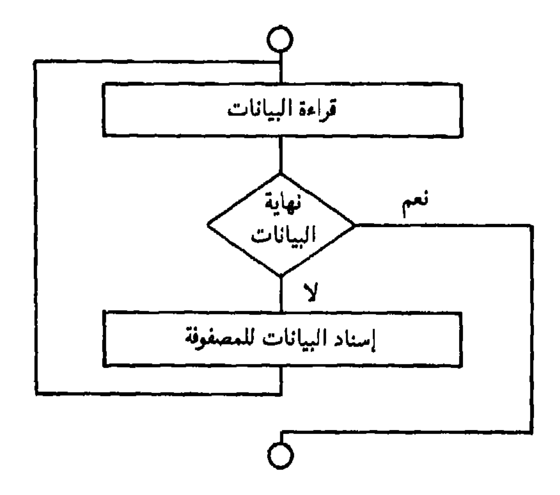
٤ ـ إيجاد متوسط درجات الحرارة للأسابيع الأ ربعة.

ثانياً ـ الهيكل الهرمى :

ثالثاً ـ نمط البرمجة الهيكلية :



قراءة درجات الحرارة ووضعها فى مصفوفة



رابعاً ــ سيتم تطوير البرنامج أولاً باستخدام تعليمات الدوارة المركبة، كما فى الشكل التالى، ومن ثم سيعدل البرنامج باستخدام تعليمة مصفوفة MAT، وذلك من أجل المقارنة بين الأسلوبين.

شكل (١٢ ــ ٢)

برنامج لإيجاد متوسط درجات الحرارة لكل أسبوع ومن ثم الأسابيع الأربعة

```
10 REM البرنامج الرئيسي
20 DIM T(4,7),SW(4)
30 GOSUB 60
40 GOSUB 120
50 END
60 REM  T برنامج فرعي لقراءة درجات الحرارة ووضعها في مصفوفة
70 FOR W = 1 TO 4:REM عدد الاسابيع
80 FOR D = 1 TO 7:REM عدد الايام
90 READ T(W,D)   :REM لقراءة درجة الحرارة لليوم
100 NEXT D :  NEXT W
110 RETURN
120 REM برنامج فرعي لايجاد متوسط درجات الحرارة لكل اسبوع
130 FOR W = 1 TO 4  :  FOR D= 1 TO 7:  S =  S  +  T (W,D)
140 SW(W) = SW(W)  +  T(W,D):     NEXT D  :  NEXT  W
150 FOR W = 1 TO 4
160 LET A = SW(W) / 7:REM لايجاد المتوسط لكل اسبوع
170 PRINT USING "##.##  =  #  متوسط درجات الحرارة لاسبوع رقم";A,W
180 NEXT W
190 REM لطباعة المتوسط العام لدرجات الحرارة
200 PRINT "_______________________________________________________"
210 PRINT  S/28; "= المتوسط العام لدرجات الحرارة في الاسابيع الاربعة  "
220 DATA 29,30,31,32,28,27,28,25,25,27,25,26,27,28
230 DATA 30,31,31,30,32,31,30,33,32,32,33,33,31,30
240 RETURN
```

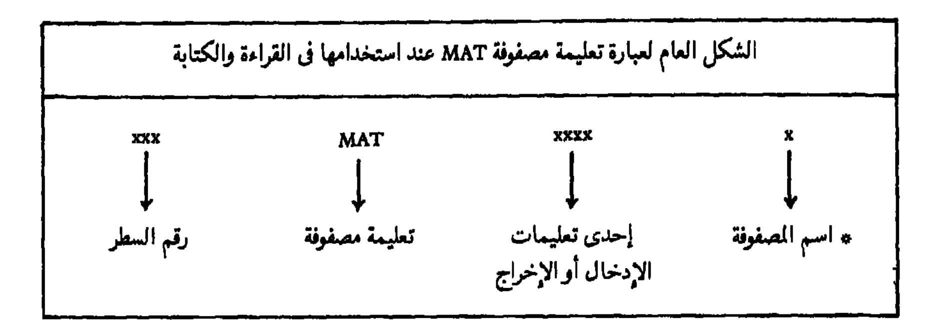
وفى حالة تنفيذ البرنامج سنحصل على النتائج التالية:

```
29.29  =  1   متوسط درجات الحرارة لاسبوع رقم
26.14  =  2   متوسط درجات الحرارة لاسبوع رقم
30.71  =  3   متوسط درجات الحرارة لاسبوع رقم
32.00  =  4   متوسط درجات الحرارة لاسبوع رقم
-------------------------------------------------------
29.53572 = المتوسط العام لدرجات الحرارة في الأسابيع الاربعة
```

وكما مر ذكره فى الجزء السابق من هذا الفصل، فإن قراءة البيانات ووضعها فى مصفوفة أو إخراج البيانات من مصفوفة، يتم بالتعامل مع كل خلية (عمود وسطر) من خلايا السطر الأول، وبعد الانتهاء منها يتم التعامل مع خلايا السطر الثانى...وهكذا...ويتم ذلك باستخدام تعليمة For - Next حيث يحدد بها عدد الأسطر والأعمدة (أبعاد المصفوفة) التى تتكون منها المصفوفة.

وباستخدام تعليمة MAT - وهى اختصار لكلمة مصفوفة MATRIX - نستغنى إلى حد كبير عن استخدام تعليمة For - Next فى التعامل مع المصفوفة.

والشكل العام لاستخدام تعليمة MAT لقراءة وطباعة المصفوفة هو كالتالى :



*** ملاحظة :**

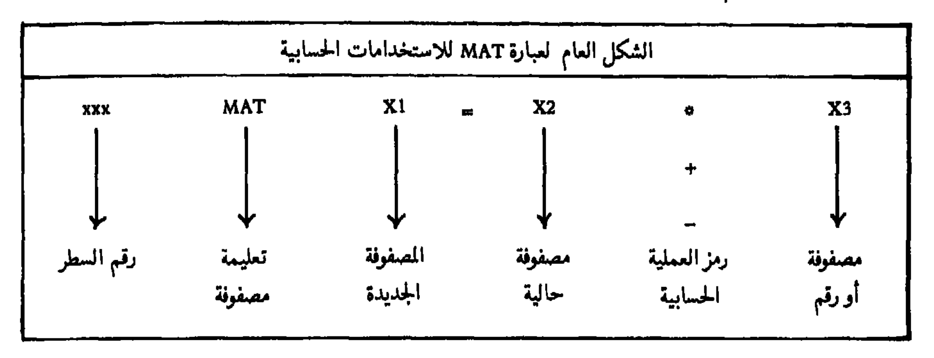
اسم المصفوفة هو الاسم الذى يلى تعليمة DIM والذى تحدد بعده (بين قوسين) أبعاد المصفوفة ، فمثلاً فى مثالنا السابق شكل (١٢ ــ ٢) فى العبارة رقم (10) والتى حددت بها أبعاد وأسماء كل من المصفوفة T وأبعادها أربعة أسطر وسبعة أعمدة والمصفوفة SW وأبعادها أربعة سطور وعمود واحد. ونستخدم هذين الاسمين T و SW للتعامل مع هاتين المصفوفتين فى حالة استخدام تعليمة MAT فلقراءة درجات الحرارة للأسابيع الأربعة نستخدم العبارة التالية : 60 MAT READ T

لتحل مكان كل من العبارات 60 إلى 90.

كما يمكننا طباعة درجات الحرارة للأسابيع الأربعة باستخدام تعليمة MAT كما هو مبين بالعبارة التالية :

```
20   MAT   PRINT   T
```

والشكل العام حسب هذا الاستعمال هو كالتالى :



فإذا أردنا تحويل درجات الحرارة من المئوية إلى الفهرنهايتية نعمل على ضرب كل خلية من خلايا المصفوفة فى القيمة $\frac{٥}{٩}$ ومن ثم نضيف القيمة ٣٢ إلى ناتج الضرب.

مقارنة بين العبارات اللازمة لتنفيذ عملية التحويل :

أ) باستخدام تعليمة FOR... NEXT

```
10 DIM C(4,7), F(4,7)
20 FOR I = 1 TO 4
30 FOR J = 1 TO 7
40 READ C (I,J)
50 LET F (I,J) = C(I,J) * (5/9) + 32
60 PRINT F (I,J)
70 NEXT J
80 PRINT
90 NEXT I
```

ب) باستخدام تعليمة MAT

```
10 DIM C(4,7), F(4,7), T(4,7)
40 MAT READ C
45 LET D = 5/9
50 MAT T = C * D
55 MAT F = T + 32
60 MAT PRINT F
```

لاحظ الفروقات مابين كل من الطريقتين (أ و ب).

١ ــ عدد العبارات التى احتجنا إليها باستخدام تعليمة FOR ... NEXT أكثر، وهذه إحدى الفوائد من استخدام تعليمة MAT حيث تعمل على تقليل عدد العبارات اللازمة لتنفيذ بعض العمليات.

٢ ــ تم التحويل من مئوية إلى فهرنهايتية بعبارة واحدة عندما استخدمنا تعليمة FOR...NEXT (عبارة 50)، وفى حالة استخدام تعليمة MAT احتجنا إلى عبارة لنسند بها ناتج قيمة $\frac{٥}{٩}$ للمتغير D (عبارة 30)، ومن ثم تمت عملية ضرب كل خلية من خلايا المصفوفة C بقيمة المتغير D وإسناد القيم إلى المصفوفة T. و بعد الإسناد أضيفت القيمة ٣٢ لكل خلية من خلايا المصفوفة T. وأسندت جميع خلاياها للمصفوفة F.

والسبب فى ذلك يعود إلى عدم إمكانية إجراء أكثر من عملية حسابية فى كل عبارة يتم استخدام تعليمة MAT بها، ولذا يتم التحايل على ذلك باستخدام مصفوفة مؤقتة، ويجب أن يكون حجم هذه المصفوفة (أى عدد السطور وعدد الأعمدة) مساوياً لحجم المصفوفة التى ستستند إليها القيمة النهائية.

٣ ــ إمكانية إجراء عملية ضرب قيمة فى كل خلية من خلايا المصفوفة باستخدام تعليمة MAT (عبارة 40).

٤ ــ إمكانية إضافة (جمع) قيمة لكل خلية من خلايا المصفوفة باستخدام MAT (عبارة 50).

العمليات الرياضية التى يمكن إجراؤها على المصفوفات باستخدام تعليمة MAT :

أولا ــ إسناد قيمة ثابتة لكل خلية من خلايا المصفوفة، فمثلاً إذا أردنا إسناد القيمة صفر (٠) لكل خلايا المصفوفة يمكننا عمل ذلك باستخدام العبارتين التاليتين :

```
10    DIM    A (2,3)
20    MAT    A = 0
```

وبعد تنفيذ العبارتين ستصبح المصفوفة A كالتالى :

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

ثانياً ــ إسناد قيم خلايا مصفوفة إلى مصفوفة أخرى :

فإذا أردنا إسناد القيم $\begin{pmatrix} 4 & 2 \\ 6 & 8 \end{pmatrix}$ للمصفوفة A

ومن ثم إسناد هذه القيم للمصفوفة B نستخدم العبارات التالية :

```
10    DIM    A(2,2), B(2,2)
20    MAT    INPUT A
30    MAT    B = A
```

وبعد تنفيذ هذه العبارات سيطلب منا إدخال قيم المصفوفة A وفى حالة إدخال القيم 2,4,6,8 فإن ناتج التنفيذ سيصبح

$$B = \begin{pmatrix} 2 & 4 \\ 6 & 8 \end{pmatrix}$$

ثالثاً ــ جمع قيم خلايا مصفوفتين، فإذا كانت لدينا المصفوفتان أ، ب

فإن عملية الجمع تتم بإضافة قيمة الخانة الأولى (سطر ١ وعمود ١) من المصفوفة الأولى إلى قيمة الخانة الأولى (سطر ١ وعمود ١) من المصفوفة الثانية وإسناد الناتج للخانة الأولى من المصفوفة الثالثة (الجديدة)، ثم يتم إضافة قيمة الخانة الثانية (سطر ١ وعمود ٢) من المصفوفة الأولى إلى قيمة الخانة الثانية من المصفوفة الثانية، وإسناد الناتج إلى قيمة الخانة الثانية من المصفوفة الثالثة، أى جمع الخلايا المتقابلة كما هو مبين فيما يلى :

C = A + B

$$\begin{bmatrix} \text{الأولى} & \text{الثانية} \\ \text{الثالثة} & \text{الرابعة} \end{bmatrix} + B \begin{bmatrix} \text{الأولى} & \text{الثانية} \\ \text{الثالثة} & \text{الرابعة} \end{bmatrix} C = \begin{bmatrix} \text{الأولى أ + الأولى ب} & \text{الثانية أ + الثانية ب} \\ \text{الثالثة أ + الثالثة ب} & \text{الرابعة أ + الرابعة ب} \end{bmatrix}$$

وإذا كانت قيمة كل من $A = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$ $B = \begin{bmatrix} 4 & 6 \\ 2 & 3 \end{bmatrix}$ فإن $C = \begin{bmatrix} 6 & 9 \\ 6 & 8 \end{bmatrix}$

و يتم ذلك باستخدام العبارات التالية :

```
10 DIM A (2,2), C (2,2)
20 MAT READ A
30 MAT READ B
40 MAT C = A + B
50 MAT PRINT A,B,C
60 DATA 2,3,4,5
70 DATA 4,6,2,3
80 END
```

و بعد تنفيذ هذه العبارات ستظهر لنا النتائج التالية :

$$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} + \begin{bmatrix} 4 & 6 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 6 & 9 \\ 6 & 8 \end{bmatrix}$$

(2+4 ، 3+6 ، 4+2 ، 5+3)

A + B = C

قيم خلايا المصفوفة A   قيم خلايا المصفوفة B   قيم خلايا المصفوفة C (ناتج عملية الجمع)

رابعاً ــ طرح قيم خلايا مصفوفة من أخرى وشروط عملية الطرح هى نفس شروط عملية الجمع أى :

١ ــ يجب أن تكون كلتا المصفوفتين متطابقتين فى الحجم (عدد الأسطر وعدد الأعمدة).

٢ ـــ يتم طرح قيمة كل خلية من المصفوفة الثانية من قيمة الخلية المقابلة لها فى المصفوفة الأولى كما هومبين فيما يلى :

C = A _ B

$$A = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \quad B = \begin{bmatrix} 4 & 6 \\ 2 & 3 \end{bmatrix} \quad C = \begin{bmatrix} 2-4 & 3-6 \\ 4-2 & 5-3 \end{bmatrix}$$

و يتم ذلك باستخدام العبارات التالية :

```
10 DIM A(2,2), C(2,2)
20 MAT READ A
30 MAT READ B
40 MAT C = A – B
50 MAT PRINT A,B,C
60 DATA 2,3,4,5
70 DATA 4,6,2,3
80 END
```

وبعد تنفيذ هذه العبارات ستظهر لنا النتائج التالية :

$$\begin{bmatrix} -2 & -3 \\ 2 & 2 \end{bmatrix} \quad \begin{bmatrix} 4 & 6 \\ 2 & 3 \end{bmatrix} \quad \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$$

قيمة خلايا المصفوفة C ناتج إعملية الطرح — قيمة خلايا المصفوفة B — قيمة خلايا المصفوفة A

خامساً ـــ ضرب قيم خلايا مصفوفتين :

إن شروط عملية الضرب التى يجب توافرها فى المصفوفات تختلف عن الشروط فى كل من عمليتى الجمع والطرح ، و يعود السبب إلى كيفية تنفيذ عملية الضرب وتحديد عدد خلايا المصفوفة الجديدة.

فالمصفوفة التى ستحتوى خلاياها على ناتج الضرب تحدد أبعادها حسب التالى :

١ ــ عدد أسطرها يكون مساو يا لعدد أسطر المصفوفة الأ ولى .

٢ ــ عدد أعمدتها يكون مساو ياً لعدد أعمدة المصفوفة الثانية .

وقبل القيام بعملية الضرب يجب تحقيق الشرط الرئيسى ، وهو أن يكون عدد أعمدة المصفوفة الأ ولى مساو ياً لعدد أسطر المصفوفة الثانية ، وإلا فلن تتحقق عملية الضرب. فمثلاً ــ إذا كانت لدينا المصفوفتان أ و ب وأبعاد كل منهما كالتالى :

المصفوفة	عدد الأسطر	عدد الأعمدة	
أ	٣	٢	لاحظ هنا أن عدد أعمدة المصفوفة أ لم تساوِ
ب	٣	٢	عدد أسطر (ب) لهذا لن تتم عملية الضرب

أما إذا كانت أبعادها كالتالى :

أ	٣	٢
ب	٢	٣

فسوف تتم عملية الضرب لأن عدد أعمدة المصفوفة أ (٢) مساوٍ لعدد أسطر المصفوفة ب أى أن الشرط الرئيسى قد تحقق، وعليه ستكون أبعاد المصفوفة الجديدة كالتالى :

عدد الأسطر = عدد أسطر المصفوفة أ أى = ٣

عدد الأعمدة = عدد أعمدة المصفوفة ب أى = ٢

ملاحظة :

بإمكاننا ضرب المصفوفة أ فى المصفوفة ب لكن لايمكننا ضرب المصفوفة ب فى المصفوفة أ لعدم تحقق الشرط الرئيسى.

فإذا كان لدينا المصفوفتان أ و ب وكانت قيمة كل منهما كالتالى :

$$\text{أ} = \begin{bmatrix} \text{د} & \text{هـ} \\ \text{و} & \text{ع} \end{bmatrix} \quad \text{و ب} = \begin{bmatrix} \text{س} & \text{ص} \\ \text{ى} & \text{ل} \end{bmatrix} \quad \text{فإن جـ} = \begin{bmatrix} \text{د×س + هـ×ى} & \text{د×ص + هـ×ل} \\ \text{و×س + ع×ى} & \text{و×ص + ع×ل} \end{bmatrix}$$

و يتم ذلك باستخدام العبارات التالية :

```
10    DIM A (2,2), B(2,2), C(2,2)
20    MAT READ A
30    MAT READ B
40    MAT C = A * B
50    MAT PRINT A,B,C
60    DATA 2,3,4,5
70    DATA 4,6,2,3
80    END
```

و بعد تنفيذ هذه العبارات ستظهر لنا النتائج التالية :

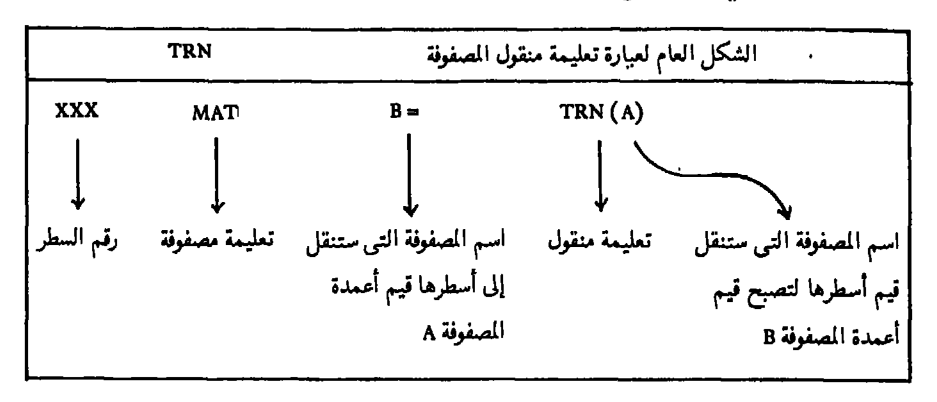
$$\begin{bmatrix} 14 & 21 \\ 26 & 39 \end{bmatrix} \qquad \begin{bmatrix} 4 & 6 \\ 2 & 3 \end{bmatrix} \qquad \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$$

قيمة خلايا المصفوفة C ناتج عملية الضرب — قيمة خلايا المصفوفة B — قيمة خلايا المصفوفة A

سادساً ــ إيجاد منقول المصفوفة TRANSPOSE OF THE MATRIX :

ومنقول المصفوفة يعنى أن قيم خلايا الأعمدة فى مصفوفة تصبح قيم خلايا السطور المقابلة لها فى مصفوفة أخرى، أى (قيم خلايا العمود الأول فى المصفوفة الأولى تصبح

قيم خلايا السطر الأول فى المصفوفة الثانية وهكذا...) و يتم ذلك باستخدام التعليمة TRN كما هو مبين بالشكل التالى :



وهنا يجب أن يكون عدد أسطر المصفوفة التى ستنقل قيم خلاياها مساو يأ لعدد أعمدة المصفوفة التى سينقل إليها.

فإذا كانت لدينا المصفوفة (أ) وقيم خلاياها كالتالى :

أ = $\begin{bmatrix} 4 & 2 \\ 3 & 5 \\ 8 & 6 \end{bmatrix}$ فإن منقولها سيكون $\begin{bmatrix} 6 & 5 & 2 \\ 8 & 3 & 4 \end{bmatrix}$

و يتم ذلك باستخدام العبارات التالية :

```
10    DIM A (3,2), B(2,3)
20    MAT READ A
30    MAT B = TRN (A)
40    MAT PRINT A,B
50    DATA 2,4,5,3,6,8
60    END
```

وبعد تنفيذ هذه العبارات ستظهر لنا النتائج التالية :

$$\begin{bmatrix} 2 & 5 & 6 \\ 4 & 3 & 8 \end{bmatrix} \qquad \begin{bmatrix} 2 & 4 \\ 5 & 3 \\ 6 & 8 \end{bmatrix}$$

قيمة خلايا المصفوفة B والتى تساوى منقول المصفوفة A   قيمة خلايا المصفوفة A

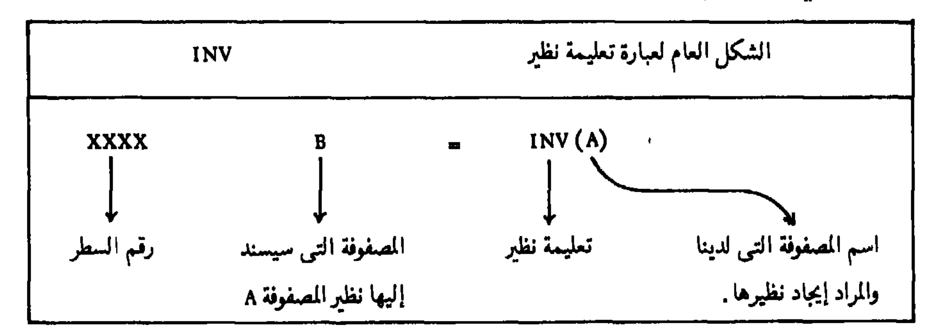
سابعاً ـــ إيجاد معكوس / نظير المصفوفة INVERSE OF THE MATRIX :

هناك العديد من الطرق والنظريات المستخدمة لإيجاد معكوس المصفوفة، ولكن جميع هذه النظريات تخرج بنتيجة واحدة وهى أن نظير المصفوفة يتمثل فى القاعدة التالية :

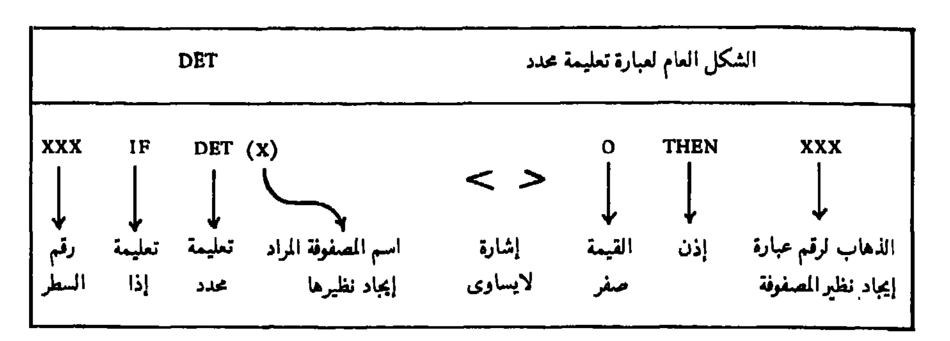
$$A * B = B * A = I$$

حيث إن I هو نظير المصفوفة الأصلية A . أما المصفوفة B فيجب أن تكون مساو ية للمصفوفة A من حيث عدد الأسطر وعدد الأعمدة، أى أن أبعادهما متساو ية . كما أنه يشترط فى المصفوفة المراد إيجاد نظيرها أن تكون أبعادها متساو ية، أى أن عدد الأعمدة يجب أن يكون مساو ياً لعدد الأسطر.

ولإيجاد نظير المصفوفة نستخدم تعليمة نظير INV وهى اختصار لكلمة INVERSE كما هو مبين بالشكل التالى :



ثامناً ـ وتستخدم تعليمة محدد DET للتأكد والاستفسار عما إذا كان يوجد لهذه المصفوفة نظير أم لا، كما هو مبين بالشكل التالى :



فإذا كانت لدينا المصفوفة A وقيمة خاناتها كالتالى

$$\begin{bmatrix} 4 & 2 \\ 3 & 1 \end{bmatrix}$$

وأردنا إيجاد نظيرها نستخدم العبارات التالية :

```
10    DIM A (2,2), B(2,2)
20    MAT READ    A
30    IF DET (A) = 0 THEN 90
40    MAT B = INV (A)
50    PRINT «المصفوفة الأصلية» : PRINT «.........»
60    MAT PRINT A : PRINT «نظير المصفوفة» : PRINT «,.........»
70    MAT PRINT B
80    GOTO 100
90    PRINT «المصفوفة فردية ولايوجد لها نظير»
100    DATA 4,2,3,1
110    END
```

فإذا قمنا بتنفيذ هذه العبارة فستكون النتائج كالتالى :

المصفوفة الأصلية	
2	4
1	3

نظير المصفوفة	
1	-0.5
-2	1.5

تطوير برنامج
تجارى باستخدام تعليمات الدوارة المركبة وتعليمة MAT :

مثال (١٢ ـ ٣) :

شركة تنتج ثلاثة أصناف باستخدام ثلاث آلات و يتم تحديد نسبة الربح المطلوبة للقطعة حسب أسلوب إنتاجها .

الهدف : إيجاد سعر البيع للوحدة المنتجة :

أولاً ـ خطوات الحل :

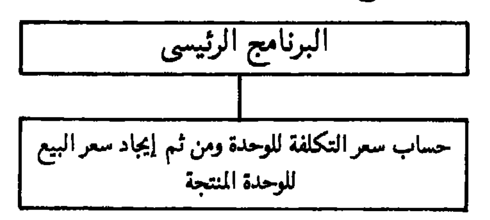
١ ـ قراءة مصفوفات التكلفة الكلية ، عدد الوحدات المنتجة ، نسبة الربح المطلوبة .

٢ ـ حساب سعر التكلفة للوحدة حيث إنه يساوى : $\frac{\text{التكلفة الكلية}}{\text{عدد الوحدات}}$

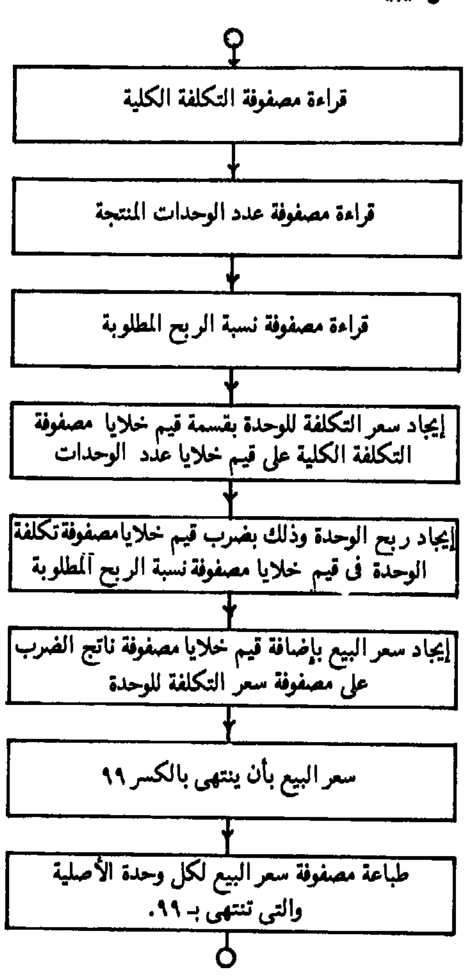
٣ ـ حساب سعر البيع = سعر التكلفة للوحدة + نسبة الربح المطلوبة بحيث تكون الأسعار XX,٩٩ .

٤ ـ طباعة النتائج .

ثانياً ــ الهيكل الهرمى للبرنامج :



ثالثاً ــ نمط البرمجة التركيبية :



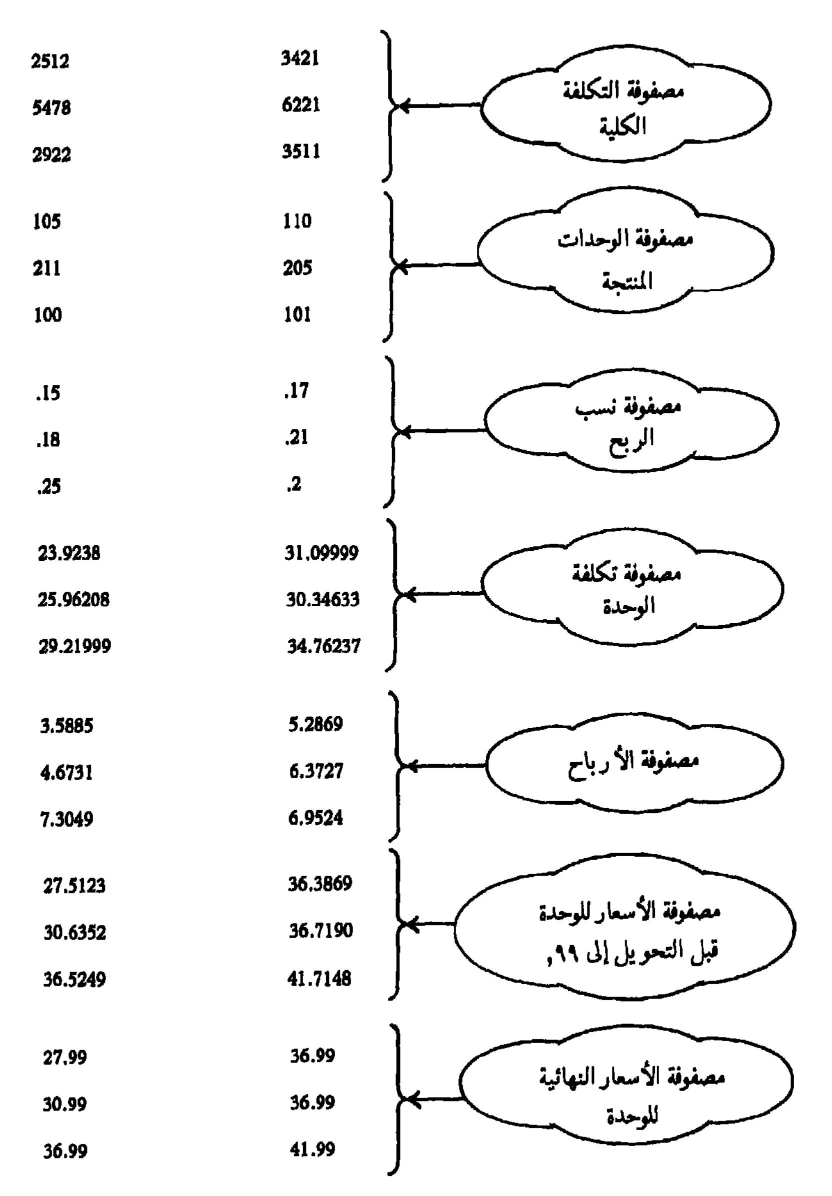
رابعاً : سيتم تنفيذ البرنامج على الحاسب الكبير IBM 3033 نظراً لعدم توفر تعليمة MAT على الحاسبات الشخصية (IBC / XT) .

شكل (١٢ ـ ٣)

برنامج لإيجاد سعر البيع للوحدة من الأصناف الثلاثة المنتجة

```
10 REM  البرنامج الرئيسي
20 REM  استدعاء برنامج فرعي
30 GOSUB 50
40 END
50 REM  برنامج فرعي للايجاد سعر البيع للوحده
60 DIM C(3,2),U(3,2),P(3,2),R(3,2),T(3,2),S(3,2),N(3,2)
70 REM  قراءة و طباعة المصفوفات الثلاثه وهم
80 REM  التكلفه الكليه , الوحدات المنتجه , نسبة الربح
90 MAT READ C,U,P
100 MAT PRINT C,U,P
110 REM  للايجاد التكلفه للوحده
120 FOR I = 1 TO 3
130 FOR J = 1 TO 2
140 LET R(I,J) = C(I,J) / U(I,J)
150 LET T(I,J) = R(I,J) * P(I,J)
160 NEXT J
170 NEXT I
180 REM  لطباعة مصفوفتي تكلفة الوحده و الا رباح
190 MAT PRINT R
200 MAT PRINT T
210 REM  للايجاد سعر البيع باضافة خلا يا ناتج الضرب على
220 REM  قيم خلا يا مصفوفة سعر التكلفه
230 MAT S = T + R
240 MAT PRINT S
250 REM  للايجاد مصفوفة الا سعار النهائيه
260 FOR I =1 TO 3
270 FOR J = 1 TO 2
280 LET N(I,J) = INT(S(I,J) + 1) - .01)
290 NEXT J
300 NEXT I
310 REM  لطباعة سعر البيع للوحده الواحده
320 PRINT
330 MAT PRINT N
340 DATA 2512,3421,5478,6221,2922,3511
350 DATA 105,110,211,205,100,101
360 DATA .15,.17,.18,.21,.25,.20
370 RETURN
```

وفى حالة تنفيذ البرنامج سنحصل على النتائج التالية :



لاحظ عدم استخدام الرسائل التوضيحية، لصعوبة تحقيق ذلك على الجهاز الكبير.

تمارين

١ ـــ اعمـل على طباعة وتنفيذ البرامج الموجودة فى أشكال الفصل، وإجراء التعديلات عليها إن وجدت، وتنفيذها حسب تسلسل هذه التعديلات.

٢ ـــ بين الأخطاء إن وجدت فى كل من العبارات التالية :

أ ـــ
```
10      DIM A (5,10), A (2,4), R
```

ب ـــ
```
10      MAT READ
```

جـ ـــ
```
10      PRINT A (I,J); (I,J)
```

د ـــ
```
10      FOR I = 9 TO 3
```

هـ ـــ
```
10      READ (I,J)
```

٣ ـــ اعمل على تعديل البرامج التالية لكى يتم تنفيذها بدون أخطاء :

أ ـــ
```
10      FOR I = 1 TO 5
20       FOR J = 1 TO 4
30      INPUT A (I,J)
40      PRINT A (I,J);
50      NEXT I
60      PRINT "________________"
70      NEXT J
80 END
```

ب ـــ
```
10      DIM A$ (4,10)
20      FOR I = 1 TO 4
30      FOR J = 1 TO 5
```

```
40   LET A$ (I,J) = I
50   NEXT J
60   NEXT I
70   END
```

جـ ــ

```
10   DIM A (3,2), B (3,2), C (2,2)
20   MAT C = A * B
```

د ــ

```
10   DIM A (10), B (5,5) A$ (20,4)
20   FOR I = 1 TO 100
30   LET B (I) = I
40   NEXT I
50   END
```

٤ ــ اشرح عمل كل من أجزاء البرامج التالية :

أ ــ

```
110   FOR I = 1 TO 2
120   FOR J = 1 TO 4
130   LET A (I) = A (I) + B (I,J)
140   NEXT J
150   NEXT I
```

ب ــ

```
110   FOR I = 1 TO 5
120   FOR J = 1 TO 7
130   PRINT A$ (I,J), A (I,J)
140   NEXT J
150   NEXT I
```

جـ ـ

```
110    DIM A (10), B (10, 10), N$ (10,10)
120    MAT INPUT A
130    MAT READ B
140    MAT READ N$
```

د ـ

```
110    DIM A(2,4), B(2,4), C(4,2), D(4,4), E(5,5)
120    MAT C = TRN (N)
130    MAT E = INV (D)
140    MAT PRINT D
```

٥ ـ اعمل على تعديل مثال (١١ ـ ١) فى الفصل الحادى عشر ليعمل على قراءة عدد الجرائد المبيعة والمعادة فى نهاية كل شهر، موزعة على الأسابيع الأربعة وأيام الأسبوع السبعة كما فى الجدول التالى :

المجموع	السبت	الأحد	الاثنين	الثلاثاء	الأربعاء	الخميس	الجمعة	الأسبوع
								١
								٢
								٣
								٤
المجموع الكلى								المجموع

المطلوب : القيام بنفس المتطلبات فى المثال.

٦ ـ يعمل ثلاثة أطباء فى إحدى العيادات الخاصة حيث يتعامل كل طبيب مع ثلاثة مستشفيات، و يتم تحويل المراجعين لأحدها حسب طبيعة تشخيص المرض. وفى نهاية كل أسبوع يقوم السكرتير بتحضير قائمة كالتالى :

رقم المستشفى	رقم الطبيب	عدد المرضى المحولين
«١» أو «٢» أو «٣»	«١» أو «٢» أو «٣»	xxx

أ ـ طور برنامجاً لقراءة البيانات المدخلة فى نهاية الأسبوع وتخزينها فى مصفوفة ذات بعدين ، ومن ثم حساب مجموع المرضى المحولين لكل طبيب ولكل مستشفى .

أمثلة من البيانات المدخلة :

1, 2,.5

2, 2, 3

1, 3, 4

3, 2, 5

ب ـ طور برنامجاً لإيجاد رقم الطبيب ورقم المستشفى اللذين لهما أقل وأكبر عدد من المرضى المحولين.

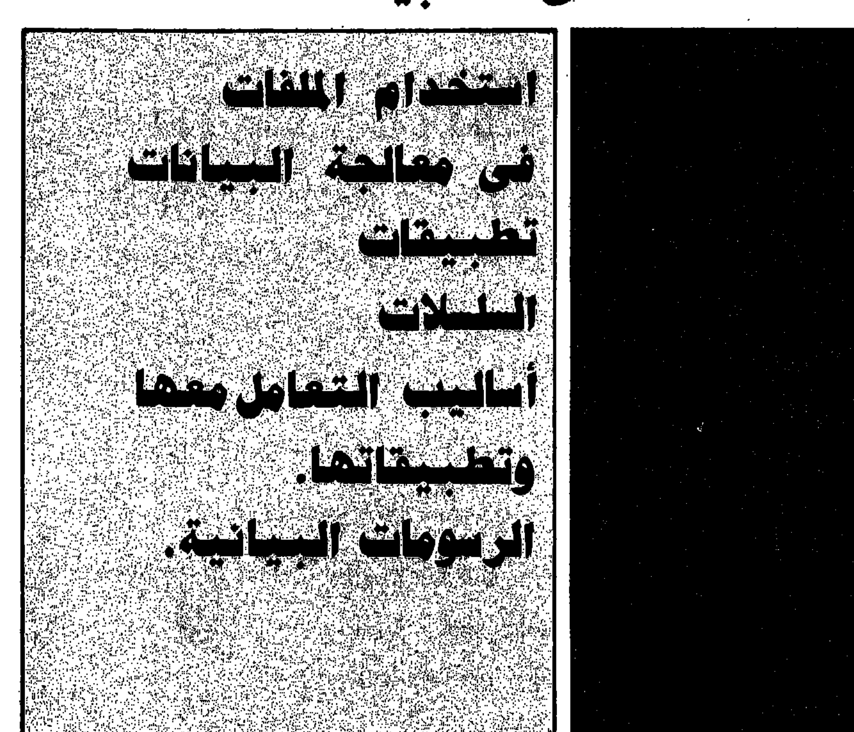
جـ ـ وإذا علم أن دخل الطبيب لكل مريض حسب المستشفى كالتالى :

رقم المستشفى	الدخل بالريالات
١	٥٠٠٠
٢	٧٠٠٠
٣	٦٠٠٠

فأوجد مجموع دخل كل طبيب من كل مستشفى.

الجزء الثالث

مواضيع متقدمة في لغة بيسك



الفصل الثالث عشر

استخدام الملفات فى معالجة البيانات

مقدمة عن معالجة الملفات :

تعرضنا فى الفصول السابقة لكيفية إدخال البيانات للحاسب ومعالجتها، ومن ثم الحصول على النتائج إما بإظهارها على الشاشة أو بالحصول على تقارير مطبوعة. ولو أردنا تكرار الحصول على هذه التقارير (نتائج المعالجة) فسنضطر لتشغيل البرنامج مرة أخرى، وفى حالة تغيير البيانات المسندة أو المرفقة مع البرنامج سنضطر لإدخالها مرة أخرى أيضا. وهذا ما يعرضنا أحياناً لبعض المشاكل، وبذل جهد ووقت إضافيين. وسنتعرض الآن لكيفية الحيلولة دون ذلك باستخدام الملفات.

- فالملف هو عبارة عن (سجل RECORD) أو أكثر من البيانات المترابطة والتى تصف شيئاً معيناً مثل الموظف أو قطع الأثاث.
- والسجل هو عبارة عن (حقل FIELD) أو أكثر من البيانات المترابطة والتى تصف آحاد أو أفراد الشيء الموصوف، مثل سجل موظف معين.
- والحقل هو عبارة عن (حرف / رقم / رمز CHARACTER) أو مجموعة من الحروف والأرقام والرموز، التى تصف حقيقة معينة عن الشيء الموصوف مثل «اسم الدارس».

فمثلاً فى قسم شؤون الموظفين يوجد ملف لجميع موظفى المؤسسة يحتوى على بيانات عن كل موظف منذ فترة التحاقه بالمؤسسة.

فنجد مثلاً : اسم الموظف، رقم الموظف، تاريخ التحاقه بالمؤسسة، المؤهل، الراتب الأساسى، علاوات، اقتطاعات...

فكل واحد من هذه البيانات يعتبر حقلاً FIELD ، ومجموع هذه الحقول يسمى السجل (سجل الموظف)، ومجموع سجلات الموظفين يطلق عليها اسم الملف.

طرق التعامل مع الملفات :

هناك طريقتان للتعامل مع الملفات : أى كتابة البيانات فى الملفات، وقراءة البيانات من الملفات، وتعديل هذه البيانات ــ وهما :

١ ــ التعامل بطريقة تتابعية : SEQUENTIAL ACCESS

وهذا يعنى قراءة أو كتابة سجلات الملف بالتسلسل : الأول فالثانى فالثالث وهكذا... إلى نهاية الملف.

ومن مميزات هذا الأسلوب أنه يصلح عند التعامل مع جميع بيانات (سجلات) الملف سواء كان هذا التعامل قراءة أو كتابة. ومن عيوبه عدم المقدرة على تعديل بيانات سجل معين على نفس الملف، إذ يتطلب ذلك إنشاء ملف جديد تسجل عليه البيانات السابقة، سواء تلك التى لم تخضع للتعديل أو تلك التى عدلت، سجلاً سجلًا، حتى نهاية الملف.

وبعد إنهاء العملية نقوم بحذف الملف القديم وتسمية الملف الجديد باسم الملف القديم. ويمكن استخدام الأشرطة والأسطوانات الممغنطة كوسائل لتخزين واسترجاع البيانات بالطريقة التتابعية.

٢ ــ التعامل بطريقة عشوائية : RANDOM ACCESS

وهى إمكانية التعامل مع أى من سجلات الملف دون المرور على مايسبق هذا السجل من سجلات أخرى. و يتم التعامل مع هذه السجلات عن طريق مفتاح KEY

معين يحدد السجل المراد. وغالباً مايكون هذا المفتاح عبارة عن أحد الحقول الرئيسية فى السجل كرقم الوظيفة مثلاً (حيث يوجد لكل موظف رقم خاص به يختلف عن رقم أى موظف آخر).

فوائد استخدام الملفات :

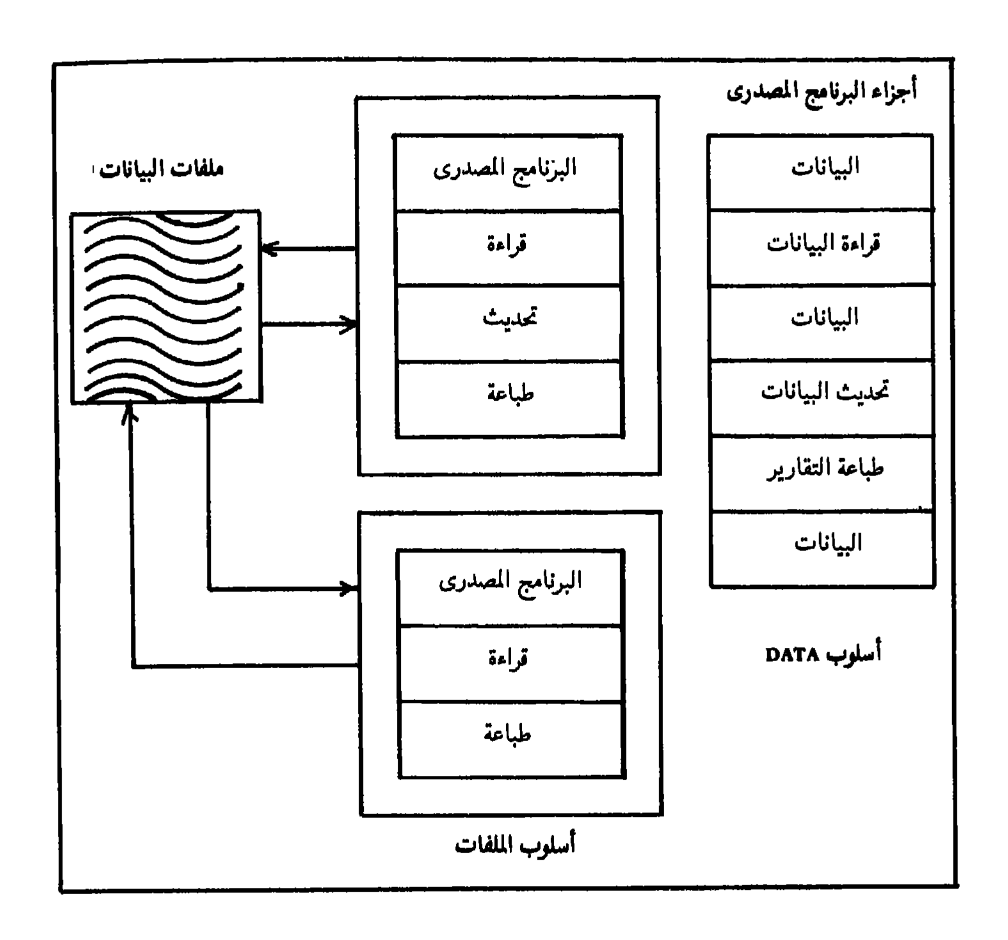
ومن الفوائد الرئيسية لاستخدام الملفات :

١) تخزين البيانات فى إحدى وسائل التخزين المستخدمة، وإمكانية الرجوع إليها فى المستقبل حين الحاجة إليها ونقلها من مكان إلى آخر.

٢) إمكانية استخدام هذه البيانات (الملف) من قبل أكثر من برنامج وذلك بالاستغناء عن تعليمة (بيانات DATA) وتوجيه البرنامج للتعامل مع الملف الذى يحتوى على البيانات المرادة.

٣) إمكانية الحصول على تقرير مطبوع عن هذه البيانات (الملف) أو إظهار البيانات على الشاشة وقت الحاجة باستخدام إحدى تعليمات أنظمة التشغيل الخاصة بذلك دون الحاجة لكتابة برنامج.

و يبين الشكل التالى الفرق بين أسلوب دمج البيانات فى البرنامج المصدرى باستخدام DATA وأسلوب الملفات.



تعليمات معالجة الملفات بالطريقة التتابعية :

وفيما يلى سنتعرض للأشكال العامة لكل من التعليمات الخاصة بتجهيز الملف والتعامل معه، قراءة وكتابة، وتحديد الانتهاء منه.

١) تجهيز وإنشاء الملف (فتح الملف) المتتابع :

OPEN المتتابع					الشكل العام لتعليمة فتح الملف	
X X X	OPEN	« XXXX »	FOR	XXXX	AS	# N
رقم السطر	تعليمة افتح	اسم الملف * ← قواعد اختيار اسم الملف	لأجل	إحدى التعليمات التالية (INPUT, OUTPUT, APPEND)	كأنه	رقم الملف

وفى حالة تنفيذ هذه العبارة سيتم إنشاء ملف يطلق عليه الاسم الموجود بين علامتى التنصيص، وهذا الإنشاء يتم فى حالة تنفيذ هذه العبارة للمرة الأولى، أما فى المرات المتتالية فيتم فتح الملف الذى اسمه يتطابق مع اسم الملف الموجود بين علامتى التنصيص، وسيستخدم هذا الملف على حسب التعليمة التى تلى تعليمة (لأجل FOR)، فمثلاً إذا كانت هذه التعليمة تساوى :

١ ـــ للإدخال INPUT فذلك يعنى أن هذا الملف سيستخدم للقراءة منه.

٢ ـــ للإخراج OUTPUT فذلك يعنى أن هذا الملف سيستخدم للتخزين (كتابة السجلات) فيه لأول مرة.

٣ ـــ للإضافة APPEND فذلك يعنى أن هذا الملف يحتوى على بيانات مخزنة و يراد إضافة بيانات (سجلات) جديدة تلى البيانات الموجودة فى الترتيب، ويمكن لهذه التعليمة أن تحل محل عبارة OUTPUT.

أما تعليمة (كأن AS) فنستخدمها لتحديد رقم تسلسل هذا الملف فى البرنامج، فهنا مثلاً تم تحديد رقم الملف بواحد 1* و يستخدم هذا الرقم للتفريق ما بين الملفات (فى

* قواعد اختيار اسم الملف :

١ ـــ يجب ألا يزيد عدد الأحرف المكون منها الاسم على تسعة أحرف.

٢ ـــ يجب أن يكون الاسم خالياً من الرموز الخاصة.

حالة وجود أكثر من ملف) فكل واحد يأخذ رقماً خاصاً به وعبارة (افتح OPEN) خاصة به، وعندما يراد استخدام أى من هذه الملفات يتم استخدام هذا الرقم حتى يتم تحديد أى من هذه الملفات هو المقصود.

٢) التعامل مع هذه الملفات، كما هو مبين فى الشكل التالى :

الشكل العام لتعليمة التعامل مع الملف (كتابة PRINT أو قراءة INPUT)				
X,((,)),X,$,((,)) ,.....	;	# N	XXXX	XXX
أسماء الحقول التى يتكون منها السجل مفصولة بفواصل لكتابتها على الملف.	فاصلة منقوطة	رقم الملف الذى حدد فى عبارة افتح OPEN	إحدى التعليمتين اكتب PRINT أو اقرأ INPUT	رقم السطر

يعتمد تركيب هذه العبارة على عاملين رئيسيين :

١ ــ نوع العملية المراد تنفيذها.

٢ ــ تركيب العبارة التى استخدمت لفتح الملف.

فإذا أردنا التعامل مع الملف للقراءة منه INPUT يجب أن تكون التعليمة المستخدمة بعد تعليمة لأجل FOR فى عبارة افتح OPEN تساوى تعليمة إدخال INPUT، وإلا سيحصل خطأ بسبب عدم تطابق نوع العملية المراد التعامل مع الملف بها مع تركيب عبارة افتح OPEN.

كما يجب أن يكون رقم الملف N # مطابقاً للرقم المستخدم فى عبارة افتح OPEN وكذلك يجب أن تكون أسماء هذه الحقول مطابقة لنوعية البيانات التى يتكون منها السجل.

٣) الانتهاء من التعامل مع الملف : يجب إغلاقه قبل إنهاء عمل البرنامج، وذلك باستخدام تعليمة أغلق كما هومبين بالشكل التالى :

الشكل العام لتعليمة أغلق CLOSE		
# N	CLOSE	XXX
↓	↓	↓
رقم الملف	تعليمة أغلق	رقم السطر

٤) يمكن الاستفسار بعد قراءة كل سجل عن نهاية الملف (نهاية البيانات) باستخدام تعليمة نهاية الملف E O F وهى اختصار لـ END OF FILE كما هومبين فى الشكل التالى :

الشكل العام لعبارة الاستفسار عن نهاية الملف E O F					
XXX	THEN	(N) *	EOF	IF	XXX
↓	↓	↓	↓	↓	↓
رقم التعليمة المراد تنفيذها فى حالة الوصول إلى نهاية الملف	إذن انتقل إلى	رقم الملف	نهاية الملف	تعليمة إذا	رقم السطر

* يجب أن يكون رقم الملف (N) مطابقاً لرقم الملف فى عبارة افتح OPEN وتستخدم هذه التعليمة فى حالة القراءة فقط.

تطوير برامج لاستخدام الملفات بالطريقة التتابعية :

مثال (١٣ ــ ١) : (الإنشاء والكتابة على الملف)

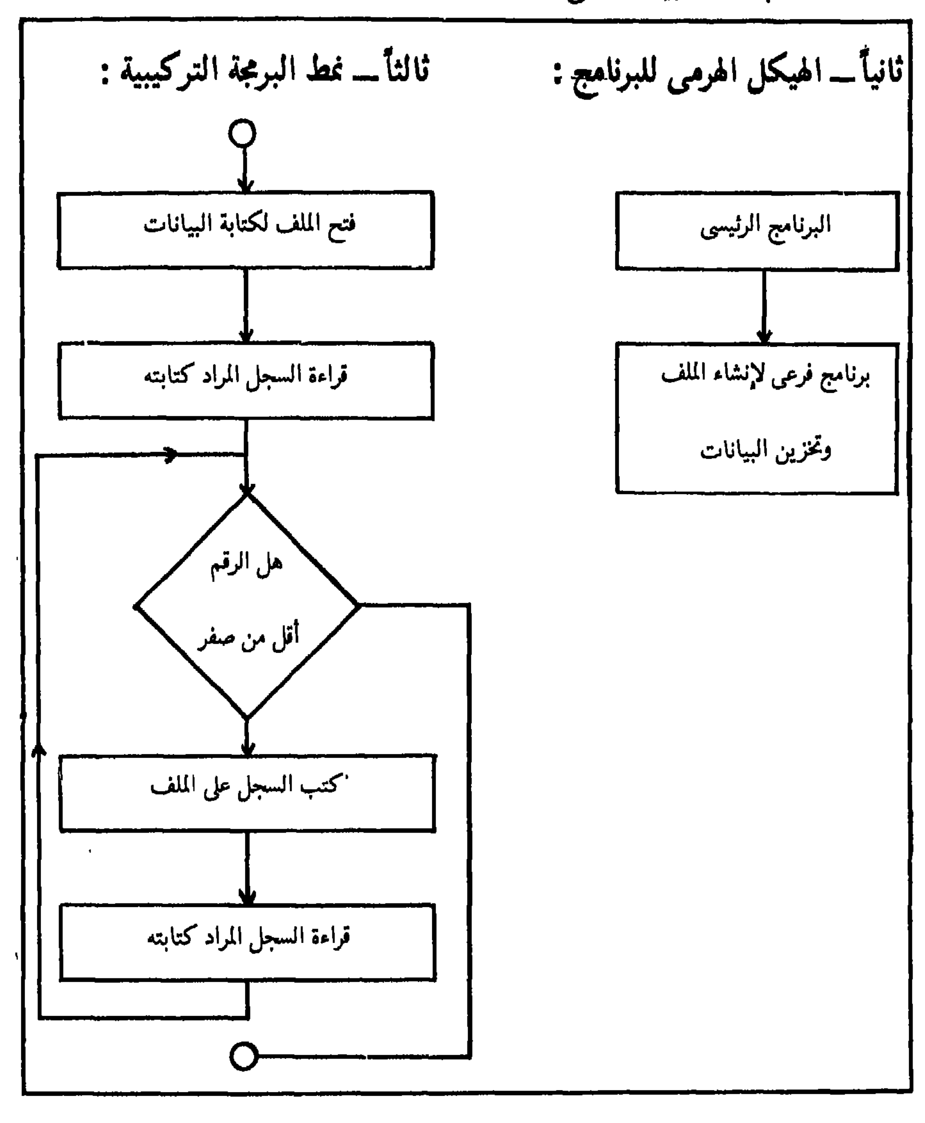
الهدف : إنشاء ملف لمخزون البضائع فى مستودع رياضى يشمل أرقام البضائع وثمن كل قطعة.

أولاً ــ خطوات الحل :

١ ــ إنشاء ملف لتخزين البيانات (أرقام القطع وسعر كل منها).

٢ ــ قراءة البيانات المراد تخزينها.

٣ ــ كتابة هذه البيانات على الملف.



رابعاً ــ وفيما يلى سنتعرض لخطوات حل المثال السابق والعبارات الخاصة بكل خطوة :

خطوة رقم ١ ــ إنشاء الملف وفتحه ، و يتم ذلك باستخدام تعليمة افتح OPEN

OPEN «PRICES» FOR APPEND AS #1

وفى حالة تنفيذ هذه العبارة سيتم إنشاء الملف وإطلاق الاسم PRICES عليه ، ولأجل تخزين/ كتابة البيانات فيه وإعطائه الرقم ١ .

خطوة رقم ٢ ــ قراءة البيانات التى ستزود مع البرنامج لتخزينها فى الملف، و يتم ذلك باستخدام تعليمة إقرأ READ و بيانات DATA

READ N , P

خطوة رقم ٣ ــ كتابة البيانات/ السجلات فى الملف ، و يتم ذلك باستخدام تعليمة اطبع PRINT

PRINT #1; N, P

وعليه سيصبح البرنامج كالتالى :

شكل (١٣ ــ ١)

برنامج لإنشاء ملف مع بيانات عن مستودع رياضى

```
10 REM استدعاء برنامج فرعي
20 GOSUB 40
30 END
40 REM برنامج فرعي لتخزين بيانات مستودع في ملف
```

```
50 REM   انشاء الملف
60 OPEN "prices" FOR APPEND AS #1
70 REM   قراءة رقم القطعه و ثمنها
80 READ N,P
90 REM   استفسار عن نهاية البيانات
100 IF N < 0 THEN 160
110 REM   كتابة البيانات على الملف
120 PRINT #1, N; ","; P
130 REM   قراءة سجل اخر
140 READ N,P
150 GOTO 100
160 REM   اغلاق الملف
170 CLOSE #1
180 RETURN
190 DATA 801,112.5
200 DATA 704, 56.2
210 DATA 620, 70
220 DATA 512, 48
230 DATA 430, 35
240 DATA 780, 65.5
250 DATA 920, 80.7
260 DATA 815, 70.5
270 DATA 720, 45.5
280 DATA -1,0
```

وبعد تنفيذ البرنامج يتم إنشاء الملف حسب البيانات المعطاة. ويمكن الحصول على نسخة من الملف باستخدام تعليمة : TYPE من وضع نظام التشغيل متبوعة بقائمة البيانات كالتالى :

```
TYPE    PRICES

801 , 112.5
704 , 56.2
620 , 70
512 , 48
430 , 35
780 , 65.5
920 , 80.7
815 , 70.5
720 , 45.5
```

مثال (١٣ ــ ٢) : (الاستفسار من الملفات)

الهدف : الاستفسار عن أثمان القطع المتوفرة فى ملف مخزون البضائع للمستودع الرياضى.

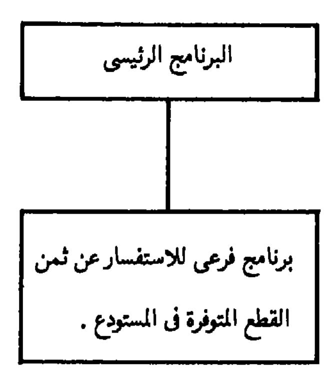
أولاً ــ خطوات الحل :

١ ــ فتح الملف .

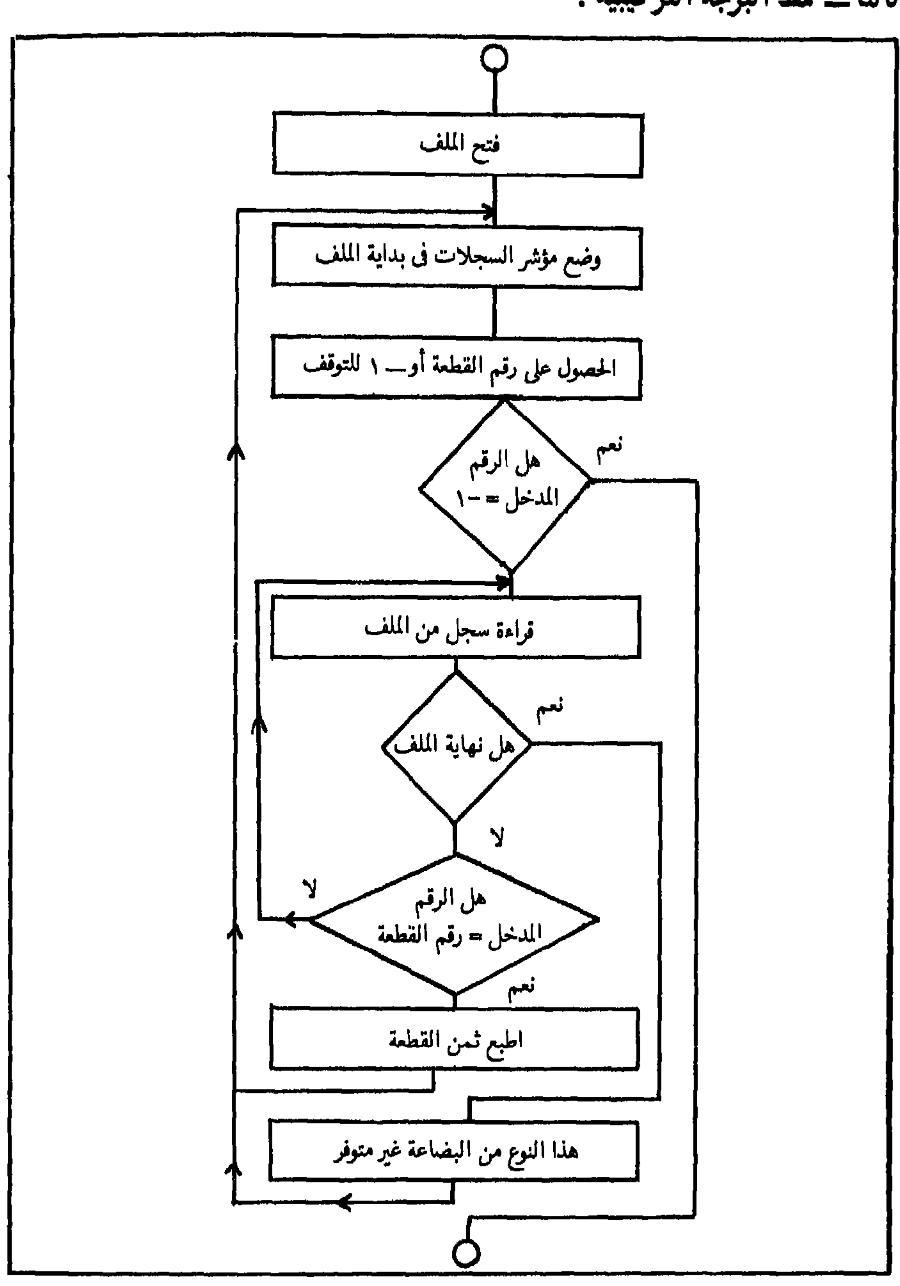
٢ ــ الحصول على رقم القطعة المراد إيجاد ثمنها .

٣ ــ قراءة سجلات الملف والمقارنة بين أرقام القطع المخزنة ورقم القطعة المدخل ، فى حالة التطابق عرض أو طباعة ثمن القطعة .

ثانياً ــ الهيكل الهرمى للبرنامج :



ثالثاً ــ نمط البرمجة التركيبية :



رابعاً ـ وفيما يلى سنتعرض لخطوات حل المثال السابق والعبارة الخاصة بكل خطوة :

الخطوة رقم ١ ـ فتح الملف و يتم ذلك باستخدام التعليمة افتح OPEN

XXX OPEN «PRICES» FOR INPUT AS #1

وفى حالة تنفيذ هذه العبارة سيتم فتح الملف الذى اسمه PRICES والذى تم إنشاؤه فى البرنامج شكل (١٣ ـ ١) ويحتوى على جميع القطع وثمن كل قطعة ، وتم تحديد الغاية من فتحه لأجل استخدامه للقراءة منه (FOR INPUT) وتم إعطاؤه الرقم ١ .

الخطوة رقم ٢ ـ الحصول على رقم القطعة المراد الاستفسار عن ثمنها ، و يتم ذلك باستخدام أى من التعليمات السابقة التى تعرفنا عليها فى كيفية إدخال البيانات للمعالجة (اقرأ READ أو أدخل INPUT ....) .

XXX INPUT ; «أدخل رقم القطعة المراد إيجاد ثمنها أو ـ ١ للتوقف» E

الخطوة رقم ٣ ـ قراءة سجلات الملف والمقارنة فى كل قراءة إذا كان رقم القطعة المخزن يساوى رقم القطعة المدخل ، اطبع ثمن القطعة ، وإلا فاستمر فى القراءة والمقارنة ، وفى حالة الوصول لنهاية الملف اعرض رسالة للمستخدم بعدم وجود هذه القطعة فى الملف ، و يتم ذلك باستخدام العبارات التالية :

XXX INPUT # 1; N,P

XXX IF EOF (1) THEN (الذهاب لعبارة عرض رسالة عدم وجود هذه القطعة)

XXX IF E = N THEN PRINT P; «ثمن القطعة =»

ELSE GOTO 30

ملاحظة :

يتم إقفال الملف بعد الانتهاء من كل عملية استفسار ، وذلك حتى يتم الرجوع إلى بداية الملف تمهيداً للقيام بالاستفسارات التالية .

وعليه سيكون البرنامج كما هو موضح فى الصفحة المقابلة.

وعند تنفيذ البرنامج للاستفسار عن أرقام القطع 620 ، 780 ، 222 ومن ثم التوقف بطباعة 1 ــ ستكون النتائج كالتالى :

====================================

620 ? أدخل رقم القطعة المراد إيجاد ثمنها أو ـ 1 للتوقف

ثمن القطعة = 70

====================================

780 ? أدخل رقم القطعة المراد إيجاد ثمنها أو -1 للتوقف

ثمن القطعة = 65.5

====================================

222 ? أدخل رقم القطعة المراد إيجاد ثمنها أو - 1 للتوقف

هذه القطعة غير موجودة فى المستودع

====================================

1 - ? أدخل رقم القطعة المراد إيجاد ثمنها أو - 1 للتوقف

================ نهاية البرنامج ================

شكل (١٣ ــ ٢)

برنامج للاستفسار عن ثمن بضائع مستودع رياضي باستخدام الملفات

```
10 REM  استدعاء برنامج فرعي
20 GOSUB 40
30 END
40 REM  لفتح الملف
60 OPEN "prices" FOR INPUT AS #1
90 INPUT "  للتوقف 1- أو  ثمنها ايجاد المراد القطعه رقم ادخل " ;E
100 REM  للاستفسار عن نهاية العمليات
110 IF E < 0 THEN 230
120 INPUT #1,N,P  : REM  للحصول على بيانات السجل
130 REM للاستفسار عن نهاية الملف
140 IF EOF (1) THEN 200
150 REM        للاستفسار عما اذا تساوي كل من رقم القع
160 IF E <> N THEN GOTO 120
170    PRINT P;"= القطعه ثمن" : CLOSE #1
180 PRINT "==========================================================":GOTO 60
190 GOTO 220
200 PRINT." المستودع في موجوده غير القطعه هذه"
210 PRINT "=========================================================="
220 CLOSE #1: GOTO 50
230 PRINT "======================  البرنامج  نهاية ======================"
240 RETURN
```

مثال (١٣ ــ ٣) : (تحديث الملفات) :

الهدف : تعديل أسعار بعض القطع المخزنة فى ملف مخزون البضائع للمستودع الرياضى .

أولاً ــ خطوات الحل :

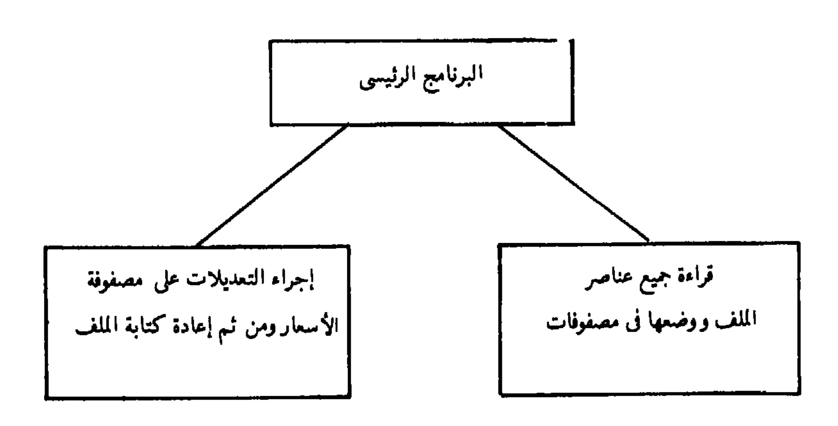
١ ــ فتح الملف وقراءة جميع السجلات ووضع البيانات فى مصفوفات طبقاً لعدد الحقول ومن ثم إغلاق الملف .

٢ ــ الحصول على رقم القطعة المراد تعديل ثمنها ، وإجراء ذلك على المصفوفات .

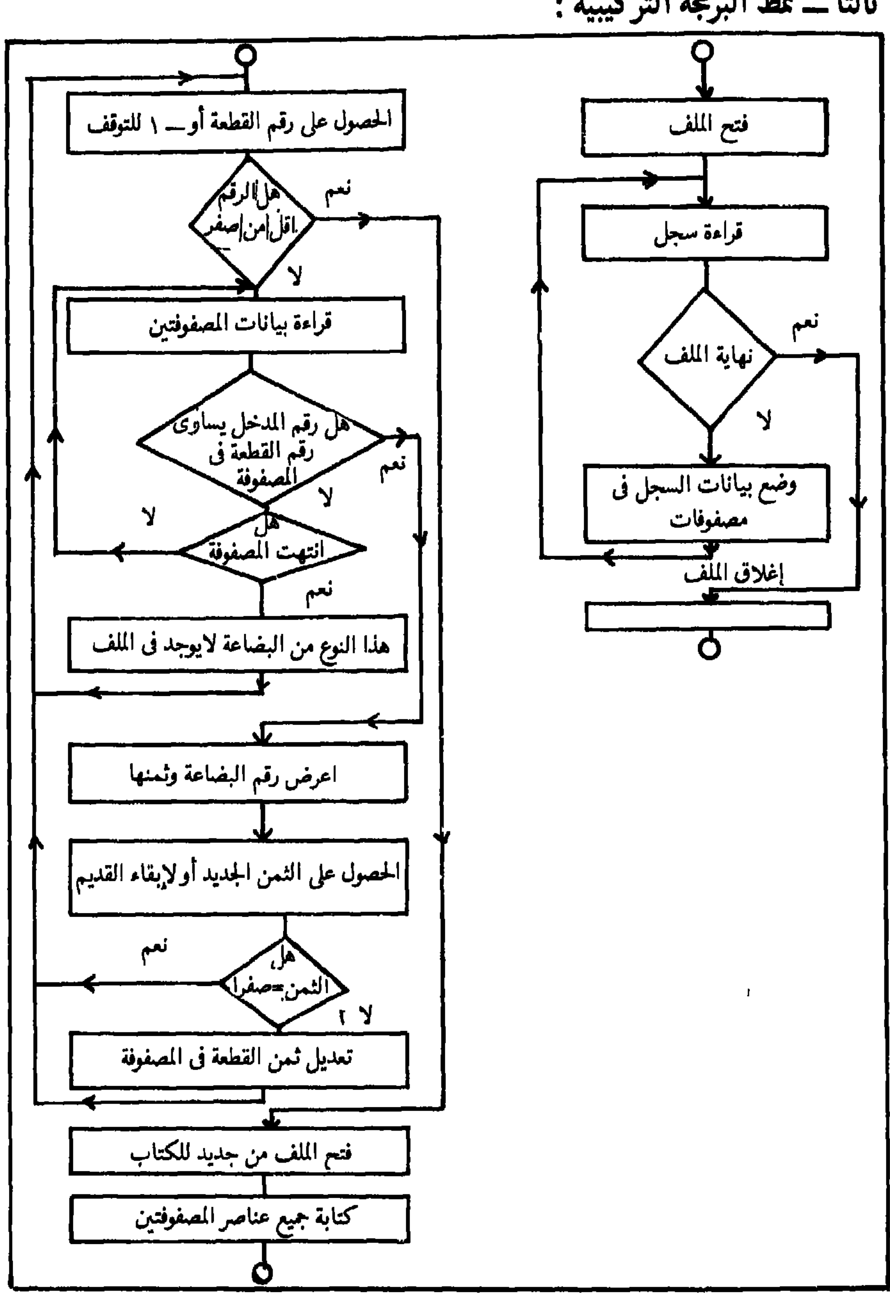
٣ ــ فتح الملف مرة أخرى وكتابة جميع عناصر المصفوفات على الملف أو إنشاء ملف جديد ، وكتابة جميع عناصر المصفوفات عليه وإعادة تسميته باسم الملف القديم .

وسنتعرض لكلتا الطريقتين فى حل المثال .

ثانياً ــ الهيكل الهرمى للبرنامج :



ثالثاً ــ نمط البرمجة التركيبية :



رابعاً ــ البرنامج كاملاً فى الشكل التالى عن إعادة كتابة البيانات على نفس الملف :

شكل (١٣ ــ ٣)

برنامج لإجراء تعديلات على سجلات الملف المتتابع للبضائع

```
10 REM  استدعاء برنامج فرعي
20 GOSUB 60
30 REM  استدعاء برنامج فرعي
40 GOSUB 140
50 END
60 REM  برنامج فرعي لقراءة بيانات الملف و وضعها في مصفوفه
70 DIM N(10),P(10)
80 OPEN "prices" FOR INPUT AS #1
90 FOR I=1 TO 10
100 INPUT #1,N(I),P(I):REM  الحصول على بيانات الملف
110 IF EOF(1) THEN 130:REM  استفسار عن نهاية الملف
120 NEXT I
130 CLOSE #1:RETURN
140 REM
150 CLS:LOCATE 10,20:PRINT "  ادخل رقم القطعه  او  1- للوقوف"
160 LOCATE 10,15:INPUT E
170 IF E <0 THEN 300
180 FOR J =1 TO I
190 IF N(J) = E THEN 240
200 NEXT J
210 REM
220 LOCATE 12,20:PRINT "لا يوجد هذا النوع من البضاعه في الملف"
```

```
230 GOTO 150
240 LOCATE 12,20:PRINT P(J);"= ثمنها الحالي";N(J)"= رقم القطعه"
250 LOCATE 14,20:PRINT "ادخل الثمن الجديد  او  0  لعدم التعديل "
260 LOCATE 14,15:INPUT C
270 IF C=0 THEN 150
280 LET P(J)=C
290 GOTO 150
300 REM
310 OPEN "prices" FOR OUTPUT AS #1
320 FOR J=1 TO I
330 PRINT #1, N(J),",",P(J):REM  كتابة البيانات الجديده
340 NEXT J
350 RETURN
```

وفى حـالـة تـنـفـيـذ الـبـرنـامـج لتعديل سعر رقم القطعة 620 من 70 إلى 85 وطباعة محتويات الملف ، ستظهر النتائج كالتالى :

أدخل رقم القطعة أو – 1 للوقوف 620 ؟
رقم القطعة = 620 ثمنها الحالى = 70
أدخل الثمن الجديد أو 0 لعدم التعديل 85 ؟

```
TYPE  PRICES
 801          ,          112.5
 704          ,          56.2
 620          ,          85
 512          ,          48
 430          ,          35
 780          ,          65.5
 920          ,          80.7
 815          ,          70.5
 720          ,          45.5
```

ويمـكـن تـعـديل البرنامج السابق لكتابة البيانات على ملف جديد ، ومن ثم إعادة تـسـمـيـة الملـف الجـديـد وإعـطـاؤه الاسم القديم وذلك بتعديل العبارات 300 إلى 350 كالتالى :

```
300 REM
310 OPEN "NEWFILE" FOR OUTPUT AS #1
320 FOR J=1 TO I
330 PRINT #1, N(J),",",P(J): REM  كتابة السجل على الملف الجديد
340 NEXT J
341 CLOSE #1
342 KILL "PRICES"
345 NAME "NEWFILE" AS "PRICES"
350 RETURN
```

ملاحظات على البرنامج السابق :

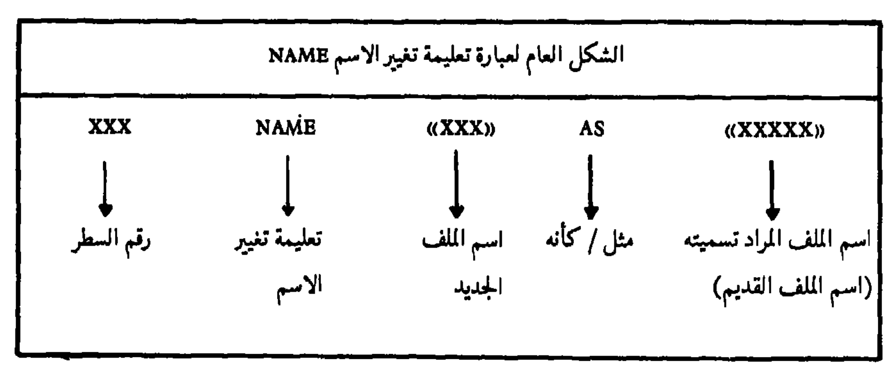
١ ـــ كما تلاحظ من شكل (١٣ ـــ ٣) ، تم تعديل الملف PRICES باستخدام المصفوفات ، ومن ثم إعادة كتابتها عليه ، أما إذا أردنا تعديل المصفوفات وكتابتها على ملف جديد ، ومن ثم تسميته باسم الملف القديم ، فيمكن عمل ذلك باتباع الخطوات التالية :

أ) تعديل العبارة رقم (310) لفتح ملف جديد بدلاً من إعادة فتح الملف وذلك بإعطاء اسم آخر للملف .

```
310 OPEN «NEWFILE» FOR OUTPUT AS ≠ 1
```

هنا سيتم إنشاء ملف جديد وإعطاؤه الاسم NEW FILE للكتابة عليه فقط .

ب) تسمية الملف الجديد (NEW FILE) بالاسم القديم (PRICES) و يتم ذلك باستخدام تعليمة (تغيير الاسم NAME) كما هو مبين بالشكل التالى :



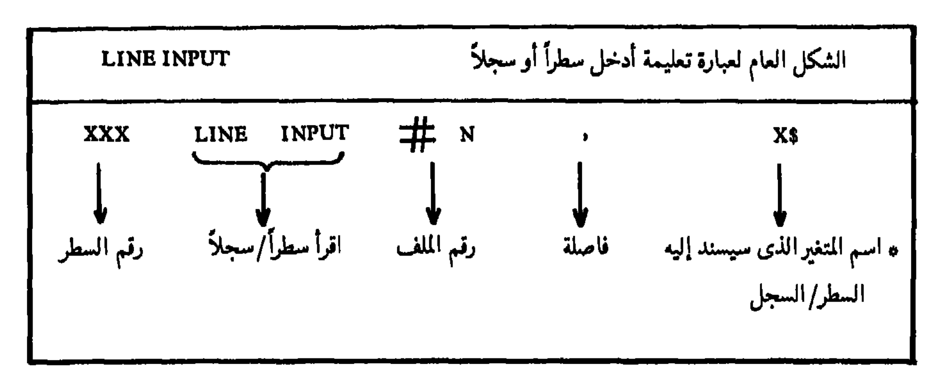
و بناء عليه نضيف العبارة التالية :

```
345 NAME «NEW FILE» AS «PRICES»
```

٢ ـــ من المستحسن فى معظم حالات التعديل القيام بعمل نسخة من الملف الرئيسى وإجراء التعديلات على النسخة الثانوية ، وتدقيق هذا التعديل فى النسخة

الثانوية؛ للحيلولة دون إجراء أى تعديل خاطىء، وبعد التأكد من صحة هذا التعديل نقوم بتسمية الملف الجديد الثانوى باسم الملف الرئيسى.

وعملية التدقيق تتم بطباعة أو عرض سجلات الملف، ويمكننا طباعة هذه السجلات سجلاً بعد الآخر دون التقيد بعدد حقول هذه السجلات بشرط ألا يزيد طول كل سجل على ٢٥٥ حرفاً/رقماً/رمزاً وذلك باستخدام تعليمة (اقرأ سطراً أو سجلاً LINE INPUT) كما هو مبين بالشكل التالى :



* يجب أن يكون اسم المتغير الذى ستسند إليه بيانات السجل غير حسابى. ورقم الملف هو الرقم الذى أسند إليه فى عبارة فتح الملف.

مثال (١٣ ــ ٤) : (طباعة ملفات)

الهدف : طباعة/ عرض بيانات (سجلات) ملف مخزون البضائع للمستودع الرياضى.

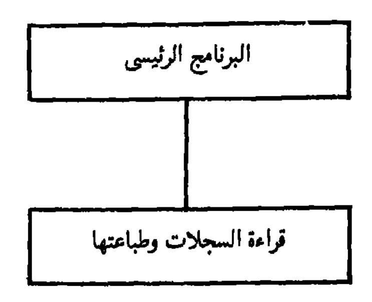
أولاً ــ خطوات الحل :

١ ــ فتح الملف.

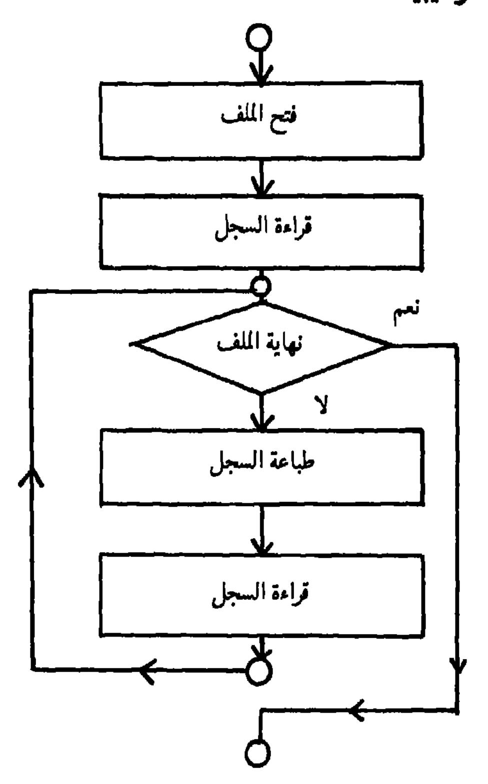
٢ ــ قراءة السجل.

٣ ــ طباعة السجل.

ثانياً ـ الهيكل الهرمى للبرنامج :



ثالثاً ـ نمط البرمجة التركيبية :



رابعاً ــ يحتوى الشكل التالى البرنامج الكامل لهذا المثال :

شكل (١٣ ــ ٤)

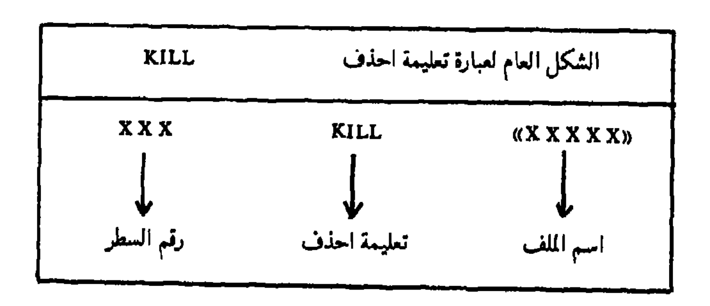
برنامج لقراءة سجلات ملف وطباعتها

```
10 REM    استدعاء برنامج فرعي
20 GOSUB 40
30 END
40 REM    برنامج فرعي لقراءة السجلات وطباعتها
50 OPEN "PRICES" FOR INPUT  AS#1:REM  فتح الملف للقراءه
60 REM    طباعة العناوين الرئيسيه لبيانات الملف
70 CLS:PRINT TAB(30);  " بيانات ملف مستودع"
80 PRINT TAB(30);"-----------------"
90 REM ( قراءة بيانات الملف (سجلا بعد الآخر
100 LINE INPUT #1,A$:REM A$    اسناد بيانات السجل للمتغير
110 IF EOF(1) THEN 140
120 PRINT TAB(30);A$:REM   طباعة بيانات السجل
130 LINE INPUT #1,A$:GOTO 110
140 PRINT TAB(30);A$:REM   طباعة بيانات اخر سجل
150 PRINT TAB(30);  "****  نهاية بيانات الملف *****"
160 RETURN
```

وفى حالة تنفيذ البرنامج السابق سنحصل على النتائج التالية :

```
بيانات ملف مستودع
-----------------
801 , 112.5
704 , 56.2
620 , 70
512 , 48
430 , 35
780 , 65.5
920 , 80.7
815 , 70.5
720 , 45.5
**** نهاية بيانات الملف *****
```

وبعد مراجعة هذه البيانات (سجلات الملف) وفى حالة عدم الحاجة لها ، يمكن الاستغناء عن الملف (أى حذفه) باستخدام تعليمة (احذف KILL) كما هو مبين فى الشكل التالى :



فمثلا إذا أردنا الاستغناء عن ملف (NEW FILE) فى برنامج شكل (١٣ — ٤) نستخدم العبارة التالية :

347 KILL «NEW FILE»

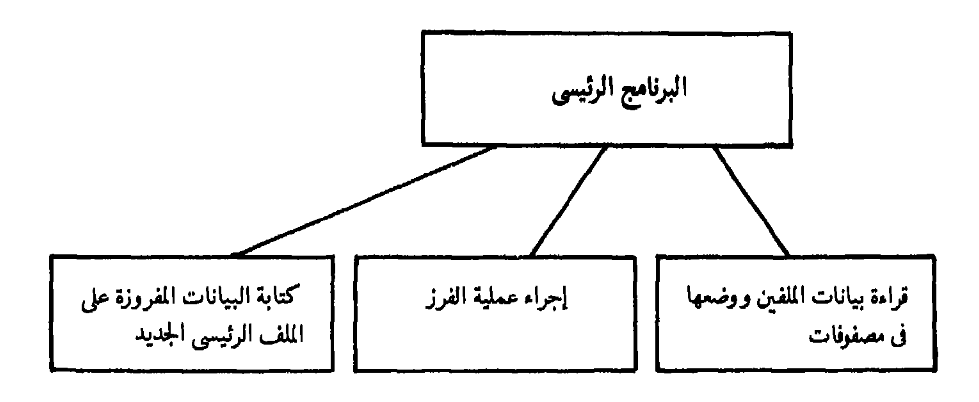
مثال (١٣ ــ ٥) : (التعامل مع أكثر من ملف) :

الهدف : إضافة بيانات مخزنة على ملف بيانات الملف الرئيسى لمخزون المستودع الرياضى (أى دمجها MERGE)، ومن ثم فرز هذا الملف الجديد وتخزين البيانات عليه مرتبة ترتيباً تصاعدياً حسب رقم الطباعة (SORTED).

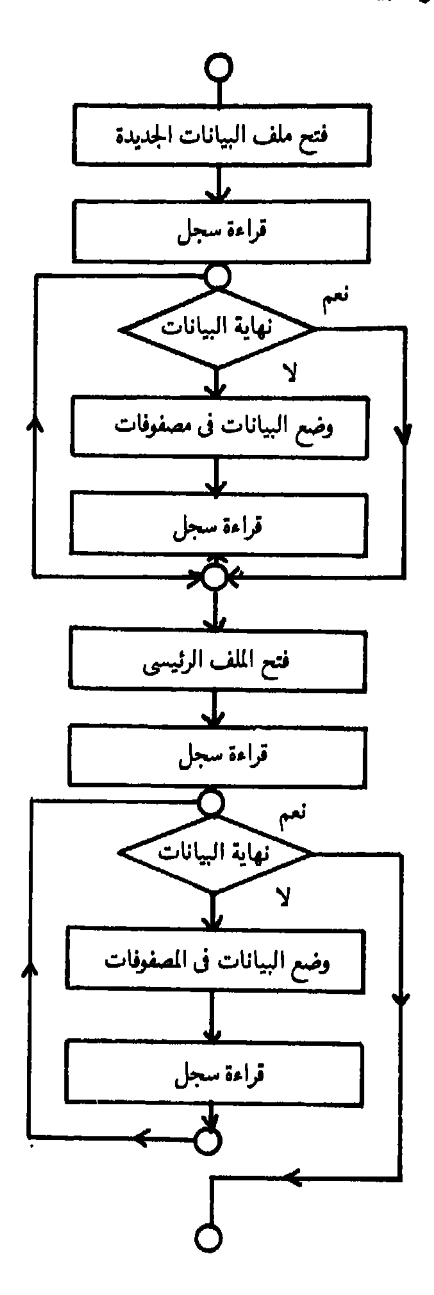
أولاً ــ خطوات الحل :

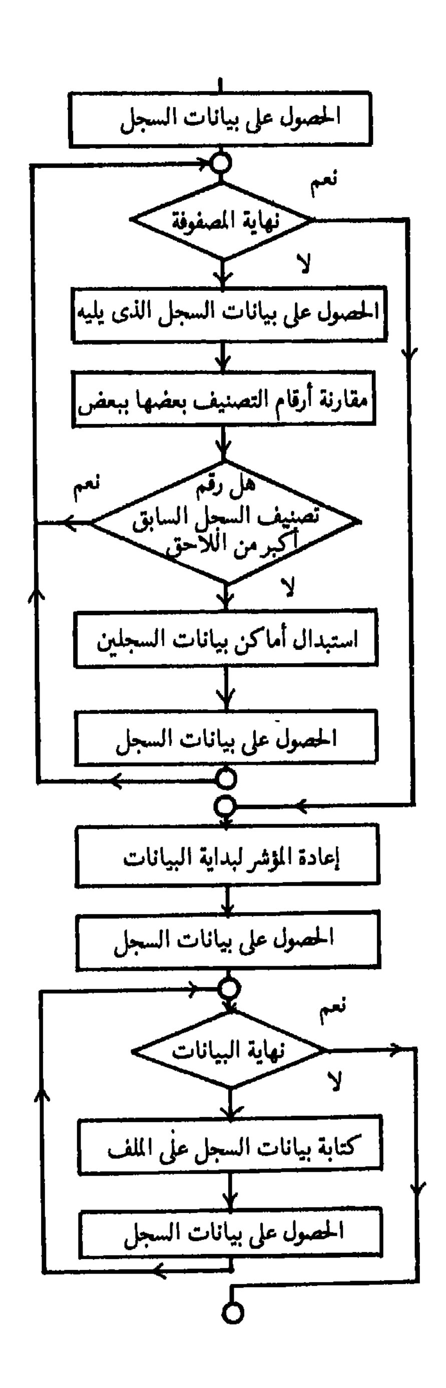
١ ــ فتح ملف البيانات الجديدة TRANSACTIONS .
٢ ــ قراءة هذه البيانات ووضعها فى مصفوفات .
٣ ــ قراءة بيانات الملف الرئيسى وإضافتها على المصفوفات .
٤ ــ إجراء عملية الفرز SORT .
٥ ــ كتابة هذه المصفوفات على الملف الرئيسى الجديد (NEW MASTER FILE) .

ثانياً ــ الهيكل الهرمى للبرنامج :



ثالثاً ــ نمط البرمجة التركيبية :




الحصول على بيانات السجل
نعم
نهاية المصفوفة
لا
الحصول على بيانات السجل الذى يليه
مقارنة أرقام التصنيف بعضها ببعض
نعم
هل رقم
تصنيف السجل السابق
أكبر من اللاحق
لا
استبدال أماكن بيانات السجلين
الحصول على بيانات السجل
إعادة المؤشر لبداية البيانات
الحصول على بيانات السجل
نعم
نهاية البيانات
لا
كتابة بيانات السجل على الملف
الحصول على بيانات السجل

رابعاً ــ البرنامج فى الشكل التالى :

شكل (١٣ ــ ٥)

برنامج لدمج ملف بيانات جديدة مع ملف البيانات الرئيسية وفرزها، ومن ثم كتابتها على الملف الرئيسى

```
10   GOSUB 50   :REM  استدعاء برنامج فرعي
20   GOSUB 200  :REM  استدعاء برنامج فرعي
30   GOSUB 310  :REM  استدعاء برنامج فرعي
40   END
50   REM "برنامج فرعي لقراءة البيانات الجديده ووضعها في مصفوفات حسب عدد الحقول"
60   DIM  N(100),P(100)
70   REM  "فتح ملف البيانات الجديد (TRANS) للقراءة منه"
80   OPEN "TRANS" FOR INPUT  AS #1
90   FOR I=1 TO 100
100  INPUT #1,N(I),P(I) :REM  لقراءة بيانات السجل ووضعها في مصفوفات
110  IF EOF (1) THEN 130 : REM  للاستفسار عن نهاية البيانات
120  NEXT I
130  REM فتح الملف الرئيسي وقراءة البيانات ووضعها في مصفوفتي الارقام والاثمان
140  OPEN "PRICES" FOR INPUT AS #2
150  FOR J=1 TO 100
160  INPUT #2,N(J+I), P(J+I):REM ( اضافة السجلات على نهاية المصفوفتين (الدمج
170  IF EOF (2) THEN 190
180  NEXT J
190  RETURN
200  REM       برنامج فرعي لاجراء عملية الفرز على البيانات تصاعديا
210  LET R=I+J :REM ايجاد عدد السجلات في الملفين
220  FOR I=1 TO R-1
```

```
230 FOR J=1 TO R-I
240 IF N(J)<= N(J+1) THEN 290
250 REM تم ايجاد رقم بضاعه اصغر من الرقم الحالي لذا سيتم استبدال الاماكن
260 LET  T=N(J):LET N(J)=N(J+1):LET N(J+1)=T
270 REM بما انه تم استبدال الاماكن للارقام لذا سيتم استبدال اسعارها ايضا
280 LET  S=P(J):LET P(J)=P(J+1):LET P(J+1)=S
290 NEXT J
300 NEXT I:RETURN
310 REM برنامج فرعي لكتابة البايانات المدموجه والمفروزه على الملف الرئيسي الجديد
320 OPEN "NSORT" FOR OUTPUT AS #3
330 FOR I =1 TO R
340 PRINT #3,N(I),",",P(I):REM كتابة السجل على الملف الجديد
350 NEXT I
360 REM لطباعة كل من ملف البايانات الجديده وبيانات الملف الجديد لمراجعتها
365 RESET
370 REM وضع المؤشر على بداية ملف البيانات الجديده
375 OPEN "TRANS" FOR INPUT AS #1
380 LPRINT TAB(10);"(TRANS ملف) البيانات المراد ادخالها
390 LINE INPUT #1,A$
400 IF EOF(1) THEN LPRINT A$: GOTO 420
410 LPRINT A$ :LINE INPUT #1,A$:GOTO 400
420 OPEN "PRICES" FOR INPUT AS #2
425 REM وضع المؤشر على بداية بيانات ملف البيانات القديمه
430  PRINT TAB(10);" (prices ملف) بيانات الملف الرئيسي القديمه"
440 LINE INPUT #2,A$
450 IF EOF(2) THEN LPRINT A$: GOTO 470
460 LPRINT A$:LINE INPUT #2,A$: GOTO 450
470 OPEN "NSORT" FOR INPUT AS #3
```

```
475 REM (newfile ملف ) وضع المؤشر على بداية البيانات المفروزه
480  PRINT TAB(10);"(newfile ملف) البيانات المدموجه و المفروزه الجديده"
490 LINE INPUT #3, A$
500 IF EOF(3) THEN 520
510  PRINT A$:LINE INPUT #3,A$: GOTO 500
520  PRINT A$:NAME "NEWFILE" AS "PRICES": KILL "NEWFILE"
530 RETURN
```

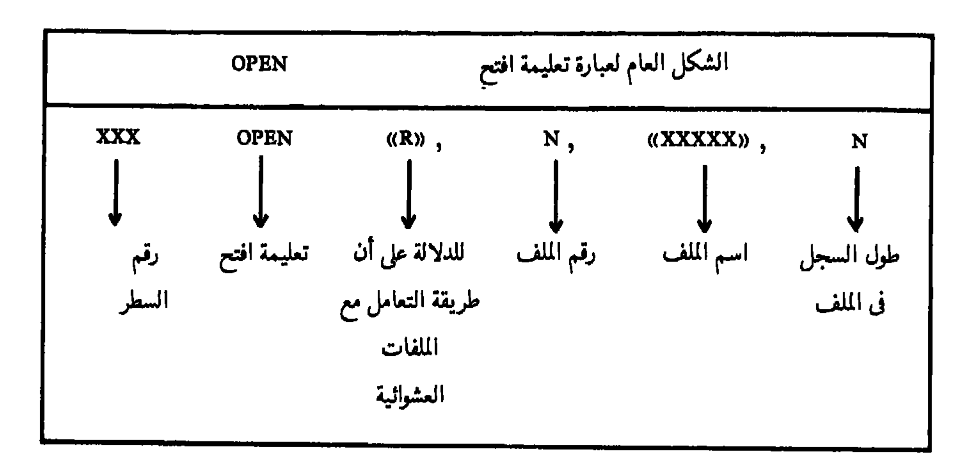
وفى حالة تنفيذ البرنامج السابق سنحصل على النتائج التالية :

```
البيانات المراد ادخالها (ملف TRANS)
777 , 55.5
999 , 66.7
818 , 70.5
735 , 45.5
بيانات الملف الرئيسي القديمه (ملف prices)
801 , 112.5
704 , 56.2
620 , 70
512 , 48
430 , 35
780 , 65.5
920 , 80.7
815 , 70.5
720 , 45.5
البيانات المدموجه و المفروزه الجديده (ملف newfile)
430          ,          35
512          ,          48
620          ,          70
704          ,          56.2
720          ,          45.5
735          ,          45.5
777          ,          55.5
780          ,          65.5
801          ,          112.5
815          ,          70.5
818          ,          70.5
920          ,          80.7
999          ,          66.7
```

تعليمات معالجة الملفات بالطريقة العشوائية :

توجد هناك اختلافات إلى حد ما فى تركيب تعليمات التعامل مع الملفات العشوائية ، عن تركيبها فى التعامل مع الملفات التتابعية ، وسنتعرض الآن لكل من هذه التعليمات :

١) **فتح الملف العشوائى :** نستخدم تعليمة افتح OPEN كما هو مبين بالشكل التالى :



لاحظ الاختلافات فى استخدام تعليمة افتح للملفات العشوائية : (١) زيادة «R» للدلالة على أن طريقة التعامل مع الملف هى عشوائية ، و (٢) طول السجل يجب تحديده . وهذا التحديد يعتمد على مجموع طول كل حقل من الحقول التى يتكون منها هذا السجل ، وإذا لم يحدد هذا الطول فسيعتبر الحاسب أن طول كل سجل من سجلات هذا الملف يساوى 255 حرفاً CHARACTER

٢) **تحديد طول السجل :** لتحديد طول السجل لابد من معرفة طول كل حقل من الحقول التى يتكون منها هذا السجل . و يتم ذلك باستخدام تعليمة الحقل FIELD كما هو مبين بالشكل التالى :

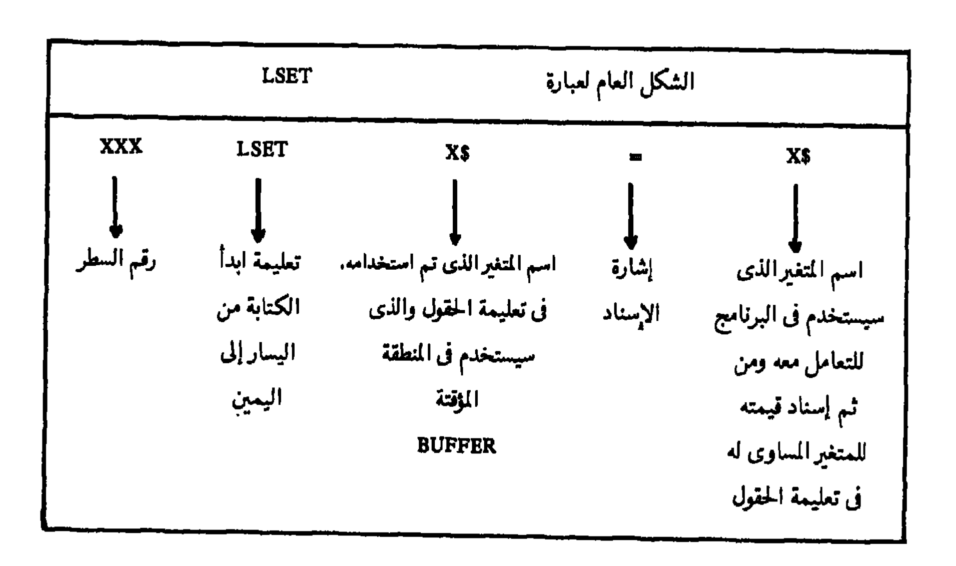
الشكل العام لعبارة تعليمة حقل FIELD								
XXX	FIELD	##N,	N	AS	X$,*	N	AS	X$...
↓	↓	↓	↓	↓	↓	↓	↓	↓
رقم السطر	تعليمة تحديد طول كل حقل من الحقول	رقم الملف	طول الحقل	كأن	اسم الحقل الأول	طول الحقل	كأن	اسم الحقل الثانى

* لاحظ أن جميع أسماء المتغيرات قد حددت كمتغيرات حرفية .

وفى حالة تنفيذ عبارة افتح OPEN فى الملفات العشوائية ، يتم حجز منطقة فى الذاكرة الرئيسية يطلق عليها اسم BUFFER (المنطقة المؤقتة) ، لاستخدامها فى حالة التعامل مع الملف (للقراءة منه أو الكتابة عليه) . فقبل أن تتم الكتابة على الملف الذى تم إنشاؤه على إحدى وسائل التخزين ، يتم وضع جميع محتو يات الحقول فى المنطقة المؤقتة BUFFER والتى تمثل السجل . و بعد ذلك تتم كتابة هذا السجل كوحدة واحدة . وما ينطبق على الكتابة ينطبق أيضاً على القراءة من الملف ، ولكن بصورة عكسية . حيث تتم قراءة السجل كوحدة واحدة و وضعه فى المنطقة المؤقتة BUFFER ، ومن ثم يمكننا التعامل مع الحقول حسب مقتضيات التطبيق .

٣) **تجهيز البيانات عقب قراءتها بواسطة تعليمة READ أو INPUT من أجل تخزينها فى الملفات العشوائية :** وتنطوى هذه الخطوة على نقل البيانات من أسماء المتغيرات التى استخدمت فى عمليات الإدخال، وتلك التى استخدمت فى تعليمة حقل FIELD . وحيث إن هناك نوعين من البيانات المدخلة ــ حرفية غير حسابية وعددية حسابية ــ فيستدعى الأمر اتباع أسلوب خاص لكل منهما .

أ ــ فالبيانات الحرفية، كالاسم مثلاً فى الأ بجدية اللا تينية، تبدأ كتابتها من اليسار إلى اليمين، ولتخزين (نقل) هذا النوع من البيانات من المتغيرات المستخدمة فى عملية الإدخال إلى متغيرات المنطقة المؤقتة BUFFER، تستعمل تعليمة LSET (أى ابدأ بكتابة البيانات فى متغير المنطقة المؤقتة من اليسار LEFT JUSTIFIED)، كما هو مبين فى الشكل التالى:



ب ــ أما فى حالة قراءة بيانات حسابية ، فيجب القيام بالخطوتين التاليتين لإتمام عملية النقل :

ــ تحويل القيمة الحسابية إلى قيمة حرفية قبل كتابتها على الملف ، وذلك باستخدام إحدى التعليمات التالية ، وفق الشكل العام التالى :

الشكل العام لعبارة تعليمة تحويل قيمة عددية إلى غير عددية				
X	XXXX$	=	X$	XXX
↓	↓	↓	↓	↓
اسم المتغير الذى سيحتوى على القيمة العددية	إحدى تعليمات التحويل للقيمة العددية إلى قيمة غير عددية	إشارة الإسناد	اسم المتغير الذى سيستخدم فى تعليمة RSET	رقم السطر

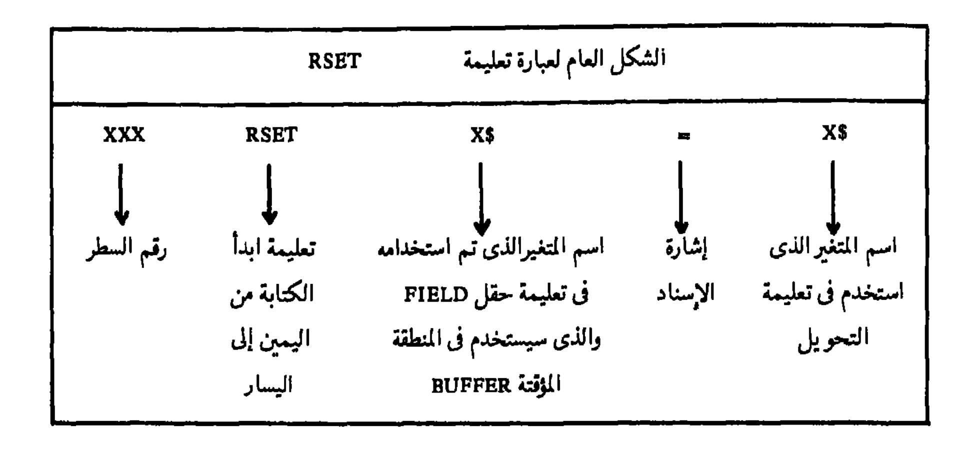
ويمكن أن تكون تعليمة التحويل إحدى التعليمات التالية :

ــ MKI$ وتعمل هذه التعليمة على تحويل القيمة العددية إلى قيمة غير عددية ، وسيتم تمثيلها فى ١٦ وحدة ثنائية أى (٢ بايت حيث إن كل بايت يساوى ثمانى وحدات ثنائية) . و يطلق عليها التعبير INTEGER EXPRESSION

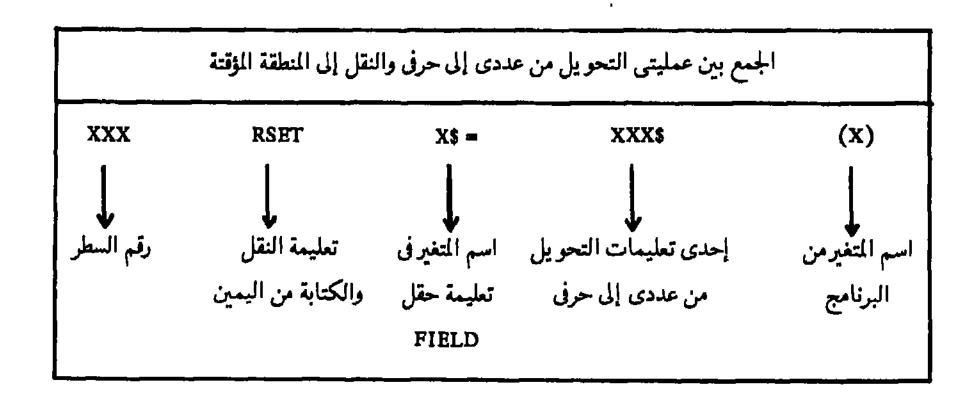
ــ MKS$ وتعمل هذه التعليمة على تحويل القيمة العددية إلى قيمة غير عددية ، وسيتم تمثيلها فى ٣٢ وحدة ثنائية أى ٤ بايت ، و يطلق عليها التعبير SINGLE-PRECISION EXPRESSION

ــ MKD$ وتعمل هذه التعليمة على تحويل القيمة العددية إلى قيمة غير عددية ، وسيتم تمثيلها فى ٦٤ وحدة أى ٨ بايت . و يطلق عليها التعبير DOUBLE-PRECISION EXPRESSION

- نقل البيانات إلى المنطقة المؤقتة وكتابتها من اليمين إلى اليسار، حيث إنها حسابية، وذلك باستخدام تعليمة RSET (أى ابدأ بكتابة الأرقام من اليمين RIGHT JUSTIFIED) كما هو مبين فى الشكل التالى :



ومن الممكن أن تجمع الخطوتان فى عبارة واحدة، كما فى الشكل التالى :



٤) الكتابة على الملف : بعد تحويل القيم المدخلة وإسنادها إلى الحقول الخاصة بها فى المنطقة المؤقتة، يمكننا كتابتها على الملف باستخدام تعليمة ضع PUT كما هو مبين بالشكل التالى :

الشكل العام لعبارة تعليمة ضع/ اكتب PUT			
X ,	# N	PUT	XXX
↓	↓	↓	↓
اسم المتغير الذى سيحتوى على أرقام سجلات الملف	رقم الملف الذى أسند إليه فى تعليمة افتح	تعليمة ضع/ اكتب	رقم السطر

فإذا كانت القيمة التى يحتويها متغير أرقام السجلات ٧، فإن ذلك يعنى أنه حال تنفيذ هذه العبارة ستتم كتابة البيانات التى تحتويها المنطقة المؤقتة BUFFER فى السجل السابع، على الملف.

٥) القراءة من الملف : أما فى حالة القراءة من الملف فتتبع الخطوات التالية :

أ ـــ إحضار السجل من الملف كوحدة واحدة، وذلك باستخدام تعليمة GET، كما هو مبين بالشكل التالى :

الشكل العام لعبارة تعليمة أحضر GET			
X	# N ,	GET	XXX
↓	↓	↓	↓
اسم المتغير الذى سيحتوى على رقم السجل فى الملف	رقم الملف الذى أسند إليه فى تعليمة افتح	تعليمة أحضر	رقم السطر

ب ــ يجب تحويل القيم العددية، والتى تم تخزينها فى صيغة قيم غير عددية، إلى صيغتها الأصلية (قيم عددية) قبل التعامل معها فى البرنامج باستخدام إحدى تعليمات التحويل، كما هو مبين بالشكل التالى :

الشكل العام لعبارة تعليمة تحويل القيم العددية والتى خزنت كغير عددية إلى أصلها عند القراءة من الملف العشوائى				
(X $)	XXX	=	X	XXX
↓	↓	↓	↓	↓
اسم المتغير الذى استخدم فى تعليمة حقل	تعليمة التحويل عند القراءة من الملف	إشارة الإسناد	اسم المتغير الذى ستسند له القيمة العددية عند التحويل والذى سيستخدم فى البرنامج	رقم السطر

ويمكن أن تكون تعليمة التحويل إحدى التعليمات التالية :

- CVI وتعمل هذه التعليمة على تحويل القيمة غير العددية (العددية الأصل) والتى تم تخزينها فى ١٦ وحدة ثنائية (٢ بايت)، وذلك باستخدام تعليمة MKI $، إلى قيمة عددية .
- CVS تحويل القيمة التى خزنت باستخدام تعليمة MKS $ إلى قيمتها الأصلية (العددية) .
- CVD تحويل القيمة التى خزنت باستخدام التعليمة MKD $ إلى قيمتها الأصلية (العددية).

أما القيم حرفية الأصل والتخزين، فيمكن استخدامها مباشرة فى البرنامج بنفس أسماء المتغيرات المستعملة فى تعليمة حقل FIELD، أو يمكن إسنادها إلى متغيرات جديدة، وذلك حسب طبيعة المعالجات .

تطوير برنامج لمعالجة الملفات بالطريقة العشوائية :

مثال (١٣ ــ ٦) : (عمليات إنشاء، إضافة، حذف، تعديل، طباعة)

الهدف : إنشاء ملف لمخزون البضائع فى مستودع رياضى يشمل جميع أرقام البضائع وثمن كل قطعة، والسماح لمستخدم النظام بالقيام بأى من العمليات التالية :

أ ــ إضافة سجل جديد .

ب ــ حذف سجل من الملف .

جـ ــ تعديل بيانات سجل .

د ــ طباعة بيانات الملف .

أولاً ــ خطوات الحل :

١) إنشاء الملف وتحديد أطوال الحقول .

٢) الحصول على نوع العملية المراد القيام بها .

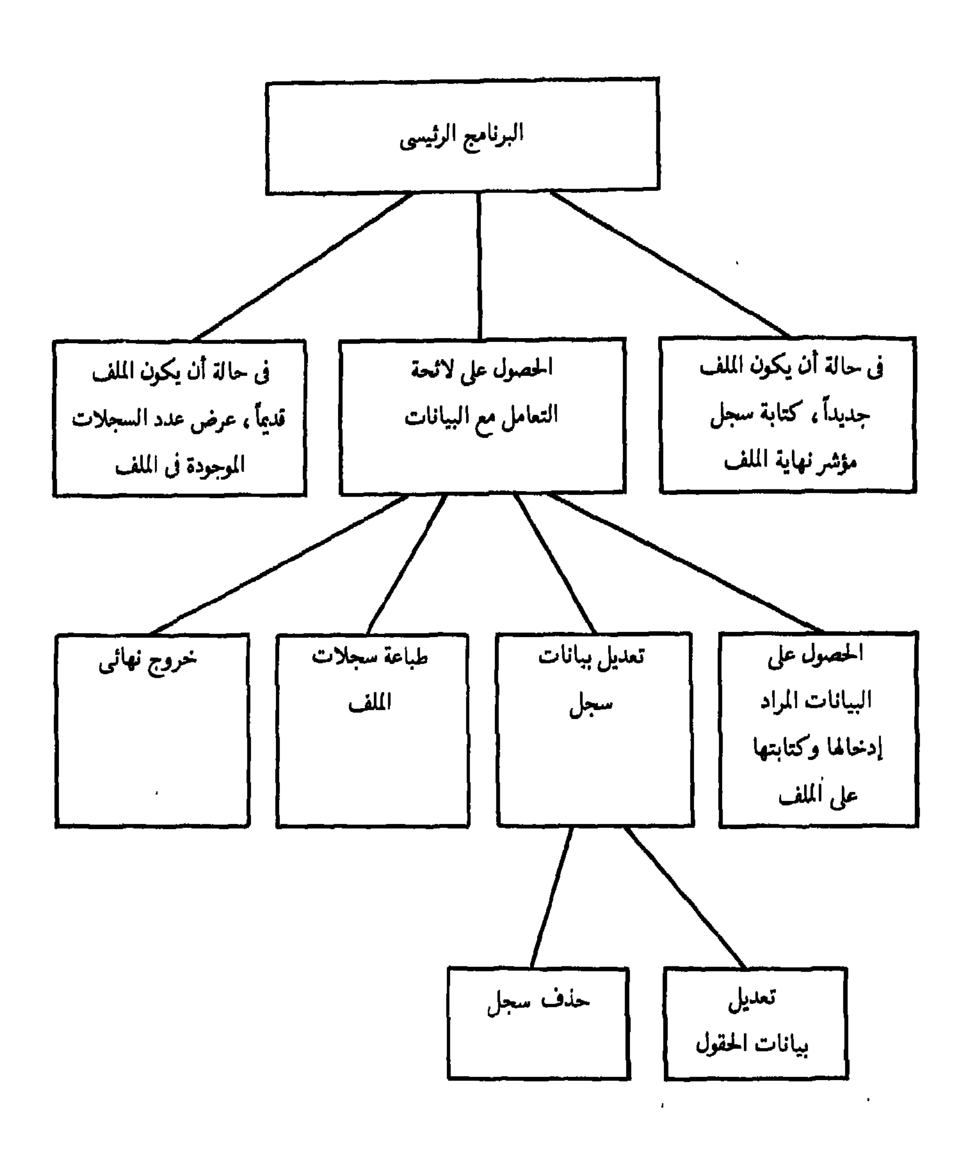
٣) فى حالة الإضافة، الحصول على البيانات، ومن ثم كتابتها على الملف .

٤) فى حالة الحذف، الحصول على السجل المراد حذفه، ومن ثم حذفه من الملف وتعديل الملف .

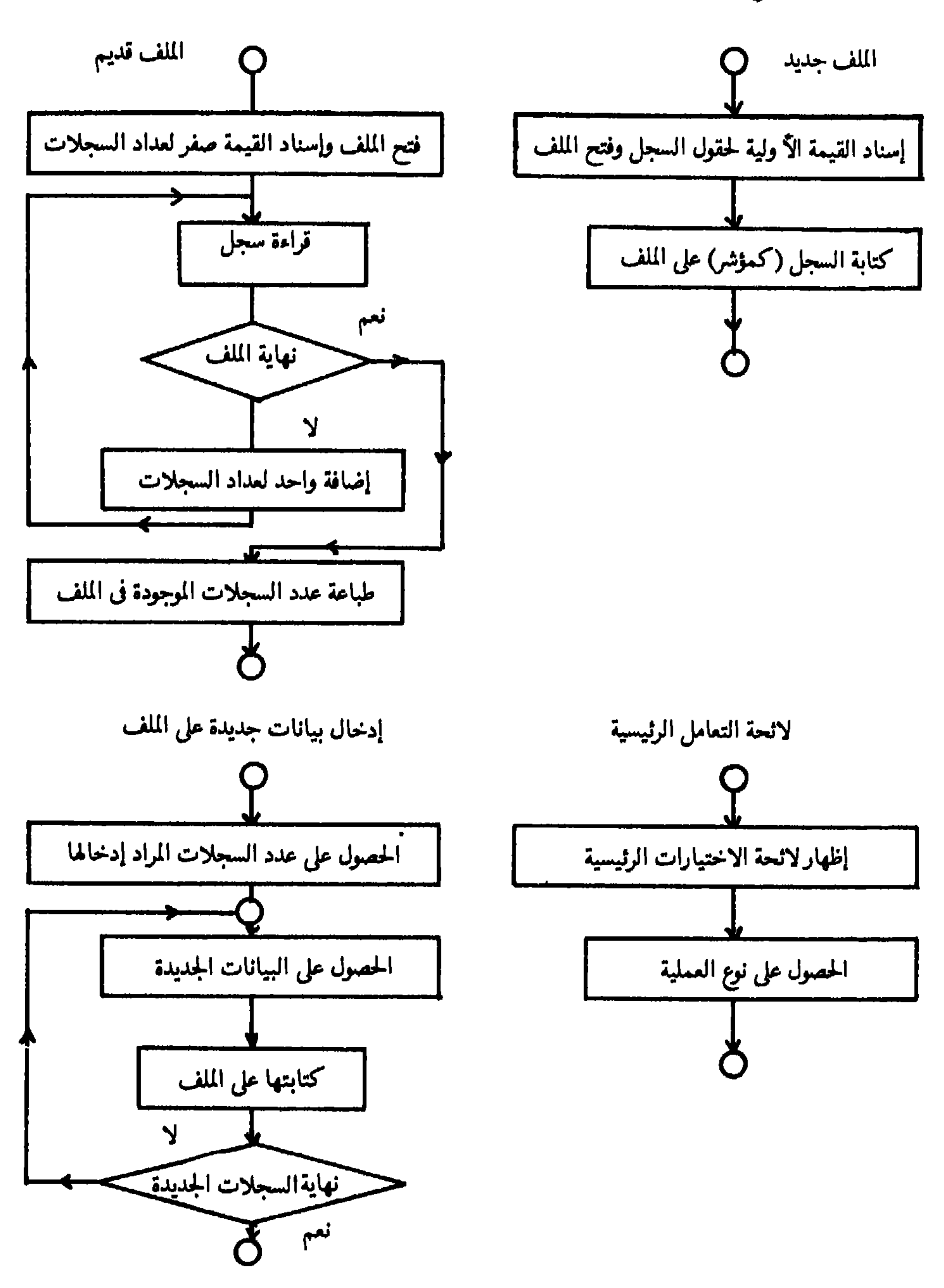
٥) فى حالة الحصول على بيانات السجل وعرضها، يتم تحديد الحقل المراد تعديله والحصول على البيانات الجديدة، وإعادة كتابة السجل على الملف .

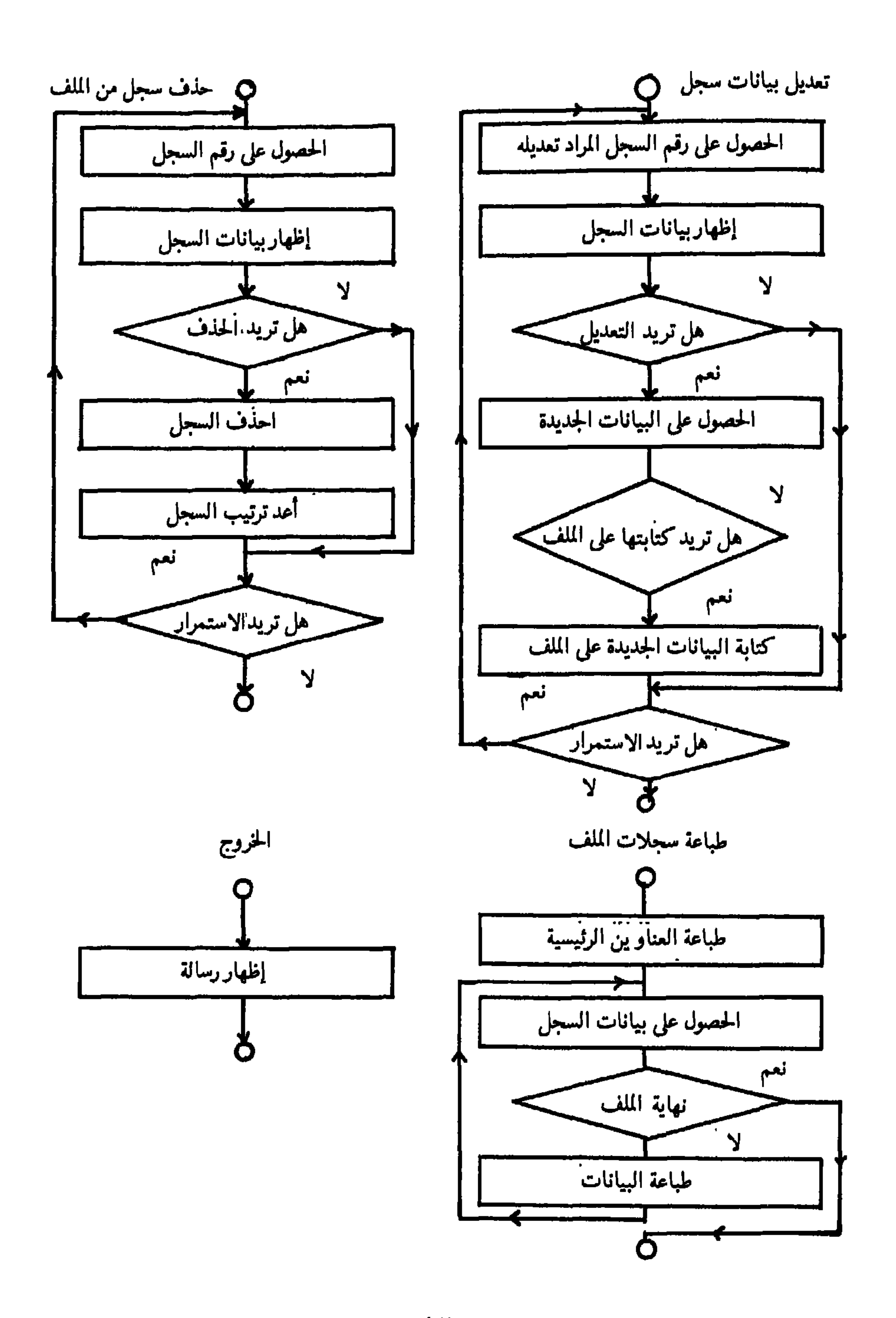
٦) فى حالة الطباعة، الحصول على جميع بيانات السجل وطباعتها .

ثانياً ــ الهيكل الهرمى للبرنامج :



ثالثاً ــ نمط البرمجة الهيكلية :





حذف سجل من الملف
الحصول على رقم السجل
إظهار بيانات السجل
هل تريد الحذف
لا
نعم
احذف السجل
أعد ترتيب السجل
نعم
هل تريد الاستمرار
لا
تعديل بيانات سجل
الحصول على رقم السجل المراد تعديله
إظهار بيانات السجل
هل تريد التعديل
لا
نعم
الحصول على البيانات الجديدة
هل تريد كتابتها على الملف
لا
نعم
كتابة البيانات الجديدة على الملف
نعم
هل تريد الاستمرار
لا
الخروج
إظهار رسالة
طباعة سجلات الملف
طباعة العناوين الرئيسية
الحصول على بيانات السجل
نهاية الملف
نعم
لا
طباعة البيانات

رابعاً ـ البرنامج كاملاً فى الشكل التالى :

شكل (١٣ ـ ٦)

برنامج لإضافة وتعديل وحذف وطباعة بيانات مستودع رياضى باستخدام الطريقة العشوائية

```
10 REM _________________ البرنامج الرئيسي _________________
20  REM     لفتح الملف العشوائى " RPRIESE " وتحديد طول السجل
30  OPEN  "RPRICES" AS #1 LEN =12
40  REM  (P$ والسعر N$ لتحديد طول كل من الحقول ( الرقم
50  FIELD #1,6 AS N$, 6 AS P$
60  REM   للاستفسار هل الملف جديد ام لا
70  CLS: LOCATE 10,20
80 PRINT "هل الملف جديد أم لا أدخل 1 = نعم أو 2 = لا "
90 LOCATE 10,20: INPUT A
100 ON A GOSUB 320,200
110 REM  لمراجعة ما تم أدخاله
120 IF A>0 AND A<=2 THEN 140
130 PRINT "========== خطا فى الادخال =======": GOTO 90
140 REM  لاستدعاء برنامج فرعى للاشحة الاختيارات الرئيسية GOSUB 500
150 GOSUB 380
160 ON T GOSUB 570,770,1250,1360
170 IF T <=3 THEN 140
180 CLOSE #1
190 END
200 REM برنامج فرعي لايجاد عدد السجلات الموجوده في الملف
210 LET R=0:CLS
220 LET R=R+1:REM  اضافة واحد لعداد سجلات الملف
```

```
230 GET #1,R :REM  للحصول على سجل
240 REM  للاستفسار عن السجل اذا كان مؤشر نهاية الملف
250 IF CVS(N$) <> 0 THEN 220
260 REM  لاظهار عدد سجلات الموجوده على الملف
270 LOCATE 10,25:PRINT R-1;"  =  عدد السجلات الموجوده في الملف"
280 LOCATE 15,25:PRINT "    اضغط على اي مفتاح للاستمرار"
290 LET I$=INKEY$
300 IF I$=""THEN 290
310 RETURN
320 REM  برنامج فرعي لانشاء الملف ووضع سجل لمؤشر نهاية البيانات فيه
330 LET N=0: LSET N$=MKS$(P)
340 LET P=0: RSET P$=MKS$(P)
350 LET R=1 :REM   عدد السجلات
360 PUT #1,R:REM  كتابة السجل لمؤشر نهاية البيانات على الملف
370 RETURN
380 REM  برنامج فرعي لاظهار لائحة الاختيارات الرئيسيه
390 CLS
400 LOCATE 5,30:PRINT "==========================================="
410 LOCATE 6,30:PRINT ":        لائحة الاختيارات الرئيسيه       "
420 LOCATE 7,30:PRINT "==========================================="
430 LOCATE 8,30:PRINT ": نوع العمليه                 الرمز      : "
440 LOCATE 9,30:PRINT ":-----------------------------------------:"
450 LOCATE 10,30:PRINT ": اضافة سجل جديد للملف        1         :"
460 LOCATE 11,30:PRINT ":          تعديل سجل          2         :"
470 LOCATE 12,30:PRINT ": طباعة جميع سجلات الملف      3         :"
480 LOCATE 13,30:PRINT ":       الجزء النهائي         4         :"
490 LOCATE 14,30:PRINT ":-----------------------------------------:"
500 LOCATE 15,30:PRINT "        : ادخل رمز العمليه المطلوبه"
510 LOCATE 15,30:INPUT T
```

```
520 REM                       لمراجعة رمز العمليه المدخله
530 IF T>0 AND T<5 THEN 560
540 LOCATE 20,17: FOR I=1 TO 5: BEEP: NEXT I
550 PRINT  "====== خطاء في الا دخال ======":GOTO 500
560 RETURN
570 REM ------------------------------------------------
580 REM              برنامج فرعي لا ضافة سجل جديد للملف
590 CLS:LOCATE 5,30 :PRINT " لا دخال بيانات سجل جديد"
600 LOCATE 10,25 :PRINT "      وثمنها:        :  ادخل رقم القطعه"
610 LOCATE 10,42 :INPUT N:LOCATE 10,20 :INPUT P
620 REM   لتحويل القيمه العدديه الىقيمه غير عدديه و أسنادهالمتغيرات المنطقه
630 LSET N$=MKS$(N)
640 LSET P$=MKS$(P)
650 PUT #1,R:LET R=R+1:REM كتابةالسجل مع الملف وزيادةعدادالسجلا ت بواحد
660 GOSUB 690: IF T2=1 THEN 590
670 LET N=0:LSET N$=MKS$(N):LET P=0:  LSET P$=MKS$(P)
680 PUT #1,R:RETURN:REM  كتابة سجل نهاية البيانات (المؤشر )والعداد
690 REM ------------------------------------------------
700 REM         برنامج فرعي للا ستفسارعن الا ستمرار في العمليه ام لا
710 LOCATE 20,30 :PRINT "      لا = 2   نعم = 1  ادخل الا ستمرار هل تريد"
720 LOCATE 20,30 :INPUT T2
730 IF T2>=1 AND T2<=2 THEN 760
740 LOCATE 21,30 :FOR I=1 TO 5 :BEEP:NEXT I
750 PRINT  "------------ خطا في الا دخال -----------":GOSUB 720
760 RETURN
770 REM ------------------------------------------------
780 REM                    برنامج فرعي لتعديل سجلا ت الملف
790 CLS :LOCATE 3,30:PRINT R-1
800 LOCATE 3,34:PRINT "    =   عدد السجلا ت الموجوده في الملف"
```

```
810 LOCATE 5,30 :PRINT  "           ادخل رقم السجل المراد تعديل بيا ناته"
820 LOCATE 5,30: INPUT R3
830 IF R3>0 AND R3<=R THEN 860 :REM  لمراجعة كلمة رقم السجل المدخل
840 LOCATE 21,20:FOR I=1 TO 5  :BEEP:NEXT I
850 PRINT "-------- خطأ في الا د خال --------":GOTO 820
860 GET #1,R3 :REM   للحصول على السجل المراد
870 CLS:LOCATE 5,20:PRINT "          وثمنها  =          رقم القطعه  ="
880 REM  لتحويل بيانات السجل من القيمه الغير عدديه الى عدديه
890 LET N=CVS(N$):LET P=CVS(P$)
900 LOCATE 5,20 :PRINT P:LOCATE 5,42:PRINT N
910 LOCATE 6,30 :PRINT "________________________________________________"
920 LOCATE 7,30 :PRINT ":          لا ئحة اختيارات التعديل          :"
930 LOCATE 8,30 :PRINT ":______________________________________________:"
940 LOCATE 9,30 :PRINT ":        نوع العمليه           الرمز        :"
950 LOCATE 10,30:PRINT ":----------------------------------------------:"
960 LOCATE 11,30:PRINT ":      حذف السجل من الملف          1       :"
970 LOCATE 12,30:PRINT ":     تعديل بيانات السجلا ت         2       :"
980 LOCATE 13,30:PRINT ":     ابقاءالبيانات كماهي           3       :"
990 LOCATE 14,30:PRINT ":______________________________________________:"
1000 LOCATE 15,30:PRINT "           ادخل رمز العمليه "
1010 LOCATE 15,30:INPUT  T4
1020 IF T4>0 AND T4<=3 THEN 1050 :REM   لمراجعةرمزالعمليه المدخله
1030 LOCATE 21,30:FOR I=1 TO 5 :BEEP:NEXT I
1040 PRINT  "---------- خطاءفي الا دخال----------":GOTO 1010
1050 IF T4=1 THEN GOSUB 1100 :REM   استدعاءبرنامج فرعي لحذف السجل
1060 IF T4=2 THEN GOSUB 1180 :REM  استدعاء برنامج فرعي لتعديل البيانات
1070 GOSUB 690 :REM  استدعاء برنامج فرعي للا ستفسار عن الا ستمرار في العمليه
1080 IF T2=1 GOTO 790 :REM     للا ستفسار عن استمرار تنفيذ العمليه
1090 RETURN
```

```
1100 REM ------------------------------------------------------------
1110 REM                       برنامج فرعي لحذف السجل من الملف
1120 GET #1,R-1 :REM             للحصول على آخر سجل تم ادخاله
1130 PUT #1,R3 :REM                 وضع المؤشرفي السجل المراد
1140 GET #1,R :REM              للحصول على آخر سجل (المؤشر)
1150 PUT #1,R-1 :REM                        تقديمه بمركز واحد
1160 LET R=R-1 :REM              طرح واحد عن عدد سجلا ت الملف
1170 RETURN
1180 REM ------------------------------------------------------------
1190 REM                        برنامج فرعي لتعديل حقول السجل
1200 LOCATE 16,30 :PRINT "-----------------------------------------"
1210 LOCATE 18,25 :PRINT "       وثمنها              ادخل رقم البضاعه"
1220 LOCATE 18,25 :INPUT P:LOCATE 18,41 :INPUT N
1230 LSET N$=MKS$(N) : LSET P$=MKS$(P) : PUT #1,R3
1240 RETURN
1250 REM ------------------------------------------------------------
1260 REM                   برنامج فرعي لطباعة محتويات الملف
1270 CLS : FOR I=1 TO R-1
1280 GET #1,I
1290 LOCATE I,10 :PRINT "       : ثمنها       : رقم البضاعه          : السجل
رقم"
1300 LET N=CVS(N$) :LET P=CVS(P$)
1310 LOCATE I,10:PRINT P:LOCATE I,26:PRINT N
1320 LOCATE I,50:PRINT I
1330 FOR W = 1 TO 8000:NEXT W:REM  للتاخير في تنفيذ البرنامج حتى تتم القراءه
1340 NEXT I
1350 RETURN
1360 REM ------------------------------------------------------------
```

```
1370 REM                         برنامج فرعي لطباعة رسالة مع السلامه
1380 CLS:FOR I=30 TO 50:LOCATE 7,I:PRINT "=";:NEXT I
1390 FOR I=7 TO 14
1400 LOCATE I,30:PRINT ":" :LOCATE I,50:PRINT ":" :NEXT I
1410 FOR I=30 TO 50:LOCATE 14,I:PRINT "=";:NEXT I
1420 LOCATE 11,31:PRINT "   مع السلامه"
1430 FOR I = 1 TO 5000:NEXT I:REM  للتاخير في تنفيذ البرنامج حتى تتم القراءه
1440 RETURN
```

وفى حالة تنفيذ البرنامج سنحصل على النتائج التالية :

أ ـ تقرير ما إذا كان السجل جديداً ، أى يكتب عليه البيانات لأول مرة .

هل الملف جديد أم لا أدخل 1 = نعم أو 2 = لا

أما إذا أدخل «2» فيتم طبع عدد السجلات فى الملف ومن ثم الاستمرار فى التنفيذ .

ب ـ إظهار لائحة الاختيارات الرئيسية :

لائحة الاختيارات الرئيسية	
الرمز	نوع العملية
1	إضافة سجل جديد للملف
2	تعديل سجل
3	طباعة جميع سجلات الملف
4	الجزء النهائى

أدخل رمز العملية المطلوبة : ؟ 1

جـ ـ إدخال البيانات (ثلاثة سجلات)

لإدخال بيانات سجل جديد

أدخل رقم القطعة : ؟ 122 وثمنها : ؟ 33

هل تريد الاستمرار أدخل 1 = نعم 2 = لا ؟ 1

لإدخال بيانات سجل جديد

أدخل رقم القطعة :   635 ؟   وثمنها :   45 ؟

هل تريد الاستمرار أدخل   1 = نعم   2 = لا   1 ؟

لإدخال بيانات سجل جديد

أدخل رقم القطعة :   817 ؟   وثمنها :   61 ؟

هل تريد الاستمرار أدخل   1 = نعم   2 = لا   2 ؟

د ـ لطباعة محتويات الملف عن طريق الاختيار رقم «3» فى الشاشة الرئيسية :

رقم السجل :	1	رقم البضاعة :	122	ثمنها :	33
رقم السجل :	2	رقم البضاعة :	635	ثمنها :	45
رقم السجل :	3	رقم البضاعة :	817	ثمنها :	61

هـ ـ تعديل سجلات الملف عن طريق إدخال الاختيار رقم «2» فى الشاشة الرئيسية :

عدد السجلات الموجودة فى الملف = 3

أدخل رقم السجل المراد تعديل بياناته 2 ؟

و ـ إظهار رقم القطعة وثمنها للبدء فى عملية التعديل، إظهار لائحة اختيارات التعديل، ومن ثم إدخال البيانات الجديدة :

رقم القطعة = 635 وثمنها = 45

لائحة اختيارات التعديل	
الرمز	نوع العملية
1	حذف السجل من الملف
2	تعديل بيانات السجلات
3	إبقاء البيانات كما هى

أدخل رمز العملية 2 ؟

أدخل رقم البضاعة 824 ؟ وثمنها 73 ؟

هل تريد الاستمرار أدخل 1 = نعم 2 = لا 2 ؟

ز ــ العودة إلى الشاشة الرئيسية وطباعة سجلات الملف بعد التعديل :

رقم السجل :	1	رقم البضاعة :	122	ثمنها :	33
رقم السجل :	2	رقم البضاعة :	824	ثمنها :	73
رقم السجل :	3	رقم البضاعة :	817	ثمنها :	61

ح ــ حذف ملف وذلك بطباعة رقم الاختيار «2» من الشاشة الرئيسية، وتحديد رقم السجل المراد حذفه، و يتم ذلك من شاشة اختيارات التعديل :

رقم القطعة = 824 وثمنها = 73

لائحة اختيارات التعديل	
الرمز	نوع العملية
1	حذف السجل من الملف
2	تعديل بيانات السجلات
3	إبقاء البيانات كما هى

أدخل رمز العملية ؟ 1

هل تريد الاستمرار أدخل 1 = نعم 2 = لا 2 ؟

ط ــ طباعة سجلات الملف بعد الحذف :

رقم السجل : 1 رقم البضاعة : 122 ثمنها : 33

رقم السجل : 2 رقم البضاعة : 817 ثمنها : 61

ى ــ للانتهاء، أدخل الاختيار رقم « » من الشاشة الرئيسية :

مع السلامة

تمارين

١ ــ اعمل على طباعة وتنفيذ البرامج الموجودة فى أشكال الفصل، وإجراء التعديلات عليها إن وجدت، وتنفيذها حسب تسلسل هذه التعديلات .

٢ ــ بين الأخطاء إن وجدت فى كل من العبارات التالية :

أ ــ
```
10   OPEN "INDATA" FOR INPUT  AS   1
```
ب ــ
```
10   LET  A$ = A
```
جـ ــ
```
10   CLOSE  INPUT
```
د ــ
```
10   FIELD ≠ 2,  B  AS  4
```
هـ ــ
```
10   OPEN R , ≠ 1, "DATA2"    20
```
و ــ
```
10   LET  B = MKI$ (B)
```
ز ــ
```
10   LET   D$ = CVI (D)
```

٣ ــ اعمل على تعديل البرنامج التالى لكى يتم تنفيذه بدون أخطاء، مع العلم بأن الملف يحتوى على بيانات عن الدارسين (رقم الدارس واسمه فقط) .

أ ــ بالطريقة التتابعية

```
10   CLS
20   LOCATE    5,80: PRINT  "         -1 للكتابة على الملف"
30   LOCATE    6, 80: PRINT  "        -2 للقراءة من الملف"
40   LOCATE    7, 80: PRINT  "        -3 لإنهاء العمليات"
50   LOCATE    8, 80: PRINT  "  ——————————————"
60   LOCATE    10,81: INPUT    A
70   ON  A  GOSB  100,200,90
80   GOTO  10
```

```
90    END
100   INPUT   "                    N$ و "أدخل رقم الدارس
110   INPUT   "                    N و " أدخل اسم الدارس
120   OPEN  "SDATA" FOR INPUT  AS  1
130   PRINT  1, N, "," N$
140   CLOSE
150   RETURN
200   CLS
210   OPEN "SDATA" FOR APPEND
220   READ  ≠  1;  N, N$
230   IF  EOF   THEN   260
240   PRINT   N$;                  رقم الدارس «N» الاسم :
250   GOTO    220
260   CLOSE  1
270   RETURN
```

ب ــ بعد إجراء التعديل وإتمام تنفيذ البرنامج السابق (أ)، اعمل على تطويره للتعامل مع الملف بالطريقة العشوائية .

٤ ــ يتابع أحد المستثمرين أسعار أسهم شركة ك ل م، وفى نهاية كل يوم يدخل السعر الجديد ويحسب نسبة التغير السلبى أو الإيجابى من سعر اليوم السابق، وفى نهاية الشهر يقوم بطبع قائمة بالأسعار بالأيام .

طور برنامجاً لهذا الغرض باستخدام الملفات، وأظهر النتائج بالأساليب التالية :

ــ قائمة بالأسعار ونسبة التغير فى كل يوم .

ملاحظة : سيحتوى الملف على حقلى السعر ونسبة التغيير.

٥ ـ يواجه محل السليمان لبيع آلات التصوير مشكلة سببها عدم حفظ سجلات دقيقة تحتوى على الأسعار الحالية للآلات ، وقد أدى ذلك إلى خسارة الزبائن بسبب بيع الآلات بأسعار أعلى من المنافسين، أو إلى تحقيق خسارة بسبب بيع الآلات بسعر أقل من سعرها الجديد .

لذلك كان من الضرورى الحفاظ على ملف للأسعار الحالية وتحديثه باستمرار، وكذلك للاستفسار عن سعر أية قطعة .

طور برنامجاً ذا ملف عشوائى للأسعار، فيه إنشاء الملف والتحديث والاستفسار .

الفصل الرابع عشر

تطبيقات

شرحنا فى الفصول السابقة كيفية تطوير برامج لتطبيقات محددة، وذلك من خلال الأمثلة المختلفة، كان آخرها فى الفصل السابق الخاص بالملفات .

وفى هذا الفصل، سنورد عدة تطبيقات ذات طابع عام وشمولى، منها العلمى والتجارى والتعليمى، وستتضمن هذه التطبيقات المجالات التالية :

١ ــ الفرز والدمج والبحث .

٢ ــ معالجة الملفات فى تطبيق علامات الدارسين الحكوميين .

٣ ــ استخدام الرسومات فى تطبيق تعليمى لعرض أجزاء الحاسب الآلى .

الفرز والدمج والبحث

Sort, Merge, and Search

تعتبر عمليات الفرز والدمج من أكثر العمليات استخداماً فى التطبيقات سواء كانت علمية أو تجارية، فنجد أن معظم أقسام المؤسسة تطلب من إدارة مركز الحاسب الحصول على تقارير مطبوعة ومرتبة ترتيباً تصاعدياً، كتقرير عن أسماء الموظفين، أو تنازليا حسب الرواتب .

وسيتم الشرح التفصيلى للخطوات المتبعة فى أساليب الفرز والبحث وإعطاء البرامج الخاصة بكل أسلوب . ومن ثم سيطور برنامج متكامل باستخدام أساليب الفرز والبحث والدمج .

الفرز : هناك العديد من النظريات المستخدمة فى عمليات الفرز ولكل منها طريقة تختلف عن الأخرى، وسنتعرض فى هذا الفصل لثلاثة من هذه الأساليب ولمزايا

وعيوب استخدام كل منها . وهذه الأساليب هي :

١ ــ أسلوب الفرز «الفقاعى» Bubble Sort

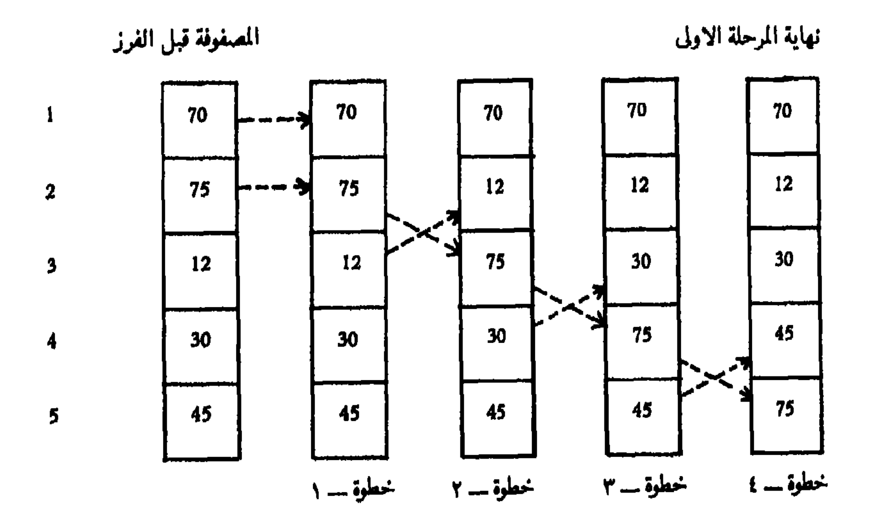
٢ ــ أسلوب الفرز «الشجرى» Heap Sort

٣ ــ أسلوب الفرز «شل» Shell Sort

مثال (١٤ ــ ١) : (أسلوب الفقاعات Bubble Sort)

باستخدام هذه النظرية تتم مقارنة كل قيمتين متجاورتين حسب نوع الفرز المراد ، فإذا كان الفرز تصاعدياً وكانت نتيجة المقارنة أن القيمة الأولى أصغر من القيمة المجاورة ، لن يتم الاستبدال والعكس صحيح فى كلتا الحالتين :

فإذا كانت لدينا المصفوفة A التالية فإن عملية الفرز التصاعدى تتم بها كالتالى :



لاحظ أنه فى نهاية المرحلة الأولى تم وضع أكبر قيمة من قيم المصفوفة فى مكانها الصحيح وهى القيمة (75) والتى يجب أن تكون فى الخلية رقم (5)

رقم الخطوة	الخلايا التى قورنت	قيمها	الناتج	الحدث
١	(1) A مع (2) A	70 مع 75	قيمة الخلية رقم ٢ أكبر	لايحدث استبدال
٢	(2) A مع (3) A	75 مع 12	قيمة الخلية رقم ٢ أكبر	يحدث استبدال
٣	(3) A مع (4) A	75 مع 30	قيمة الخلية رقم ٣ أكبر	يحدث استبدال
٤	(4) A مع (5) A	75 مع 45	قيمة الخلية رقم ٤ أكبر	يحدث استبدال

لقد احتجنا إلى أربع خطوات فى المصفوفة التى عدد خلاياها يساوى خمساً لوضع قيمة واحدة فى موضعها الصحيح . ولوضع القيمة الثانية فى الموضع الصحيح لما نبحث فى أربع خلايا بدلاً من خمس، وهكذا إلى أن يتم وضع جميع عناصر المصفوفة فى الترتيب المطلوب .

والبرنامج التالى يبين هذا الأسلوب فى الفرز.

شكل (١٤ ــ ١)

برنامج للحصول على أرقام عشوائية باستخدام دالة RND وفرزها باستخدام أسلوب BUBBLE

```
10 DIM A(1000)
20 GOSUB 50:REM استدعاءبرنامج فرعي للحصول عللا رقام عشوائية
30 GOSUB 140:REM استدعاءبرنامج فرعي لا جراءعملية الفرز
40 END
50 REM rnd برنامج فرعي للحصول على ارقام عشوائية باستخدام دالة
60 CLS:LOCATE 10,20:PRINT "ادخل عدد الا رقام المراد الحصول عليها :"
70 LOCATE 10,20 :INPUT N
80 FOR I=1 TO N
90 LET B =RND
100 REM 1000,1 تحويل الرقم العشوائي الى رقم صحيح يقع مابين
110 LET A(I)=INT(B*1000)
120 NEXT I
130 RETURN
140 REM BUBBLE برنامج فرعي لا جراء عملية الفرز باستخدام نظرية
150 LET S$=TIME$: REM لتحديد بداية وقت الفرز
160 LET M=N :REM M اسناد عدد الا رقام للمتغير
170 LET P=0: REM اسناد القيمة صفر لمؤشر الفرز
180 FOR I=1 TO M-1
190 REM لا جراء عملية المقارنة بين القيم
200 IF A(I+1)>=A(I) THEN 260
210 REM لم يتحقق الشرط فستتم عملية تغيير اماكن القيم
```

```
220 LET T=A(I+1)
230 LET A(I+1)=A(I)
240 LET A(I)=T
250 REM     اسناد قيمة I للمؤشر
260 LET P=I
270 NEXT I
280 REM للاستفسار عما اذا تم الفرز نهائيا
290 IF P=0 THEN 330
300 REM (لم يتحقق الشرط (اي لم ينتهي الفرز
310 LET M=P
320 GOTO 170
330 REM  للحصول على وقت نهاية الفرز
340 LET F$=TIME$
350 LOCATE 13,20
360 PRINT " باستخدام نظرية BUBBLE ثم فرز ";N;"من الارقام "
370 LOCATE 15,20:PRINT F$;" الى ";S$;" الوقت من
380 RETURN
```

يعتبر أسلوب الفرز بالطريقة الفقاعية من أكثر الأساليب ملاءمة لفرز البيانات قليلة العدد، ومن أسهل الأساليب تفهماً واستيعاباً خاصة للمبتدئين، لكن فى بعض الحالات نحتاج لاستخدام أساليب أخرى فى عملية الفرز، حتى ولو احتجنا إلى كتابة عبارات أكثر مما احتجنا فى أسلوب الفرز الفقاعى .

مثال (١٤ ــ ٢) : (أسلوب شل Shell)، و يطلق عليه اسم Shell نسبة إلى مكتشفه D. L. Shell فى عام ١٩٥٩م[1] .

والخطوات التى يمر بها هذا الأسلوب كالتالى :

١ ــ قراءة البيانات ووضعها فى مصفوفة والحصول على عدد العناصر N

٢ ــ قسمة المصفوفة إلى نصفين M = INT (N / 2)

٣ ــ الاستفسار إذا كان المنتصف يساوى صفراً أى أنه قد تم فرز المصفوفة، وإلا تنفذ الخطوات من ٤ إلى ٦ .

٤ ــ تحديد بداية النصف وذلك بطرح المنتصف (M) من عدد عناصر المصفوفة (N) .

S = N - M

٥ ــ تحديد نهاية النصف .

٦ ــ العودة للخطوة رقم ٢ .

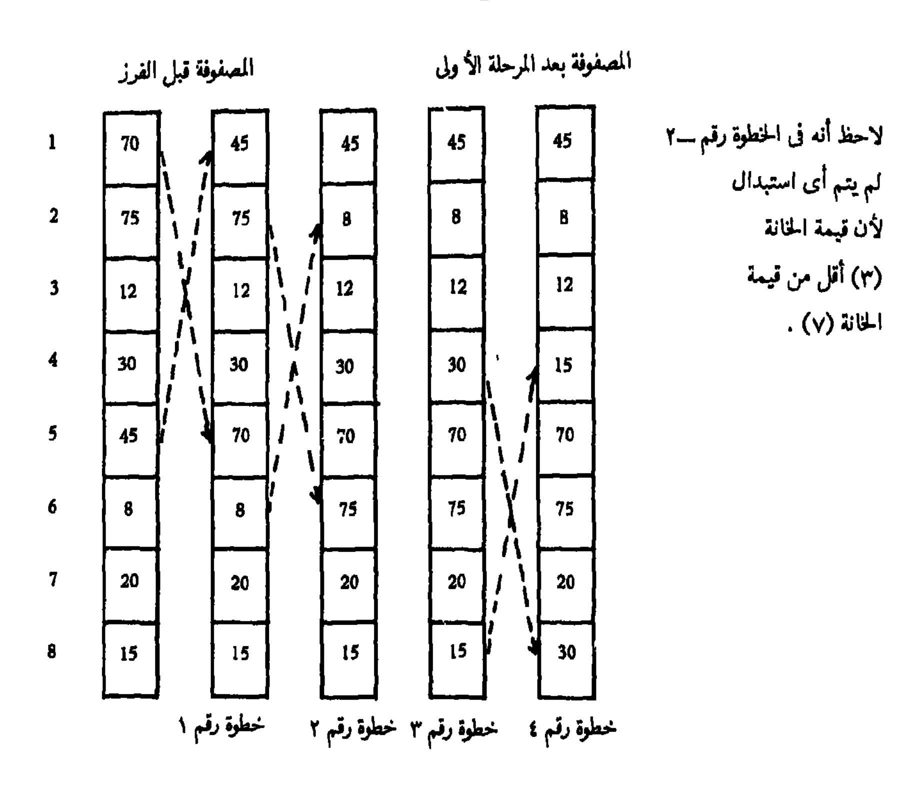
وخطوات الحل باستخدام هذا الأسلوب تتم بقسمة المصفوفة إلى قسمين، وذلك بالحصول على ناتج (العدد الصحيح) من قسمة عدد خلايا المصفوفة على اثنين، ويمثل هذا الناتج المسافة (عدد الخانات) بين كل خانتين ستتم المقارنة بين قيمهما، و يطلق عليه اسم الفراغ .

---

(1) A High-Speed Sorting Algorithm, Communications of the ACM, July 1959, Vol. 2,PP. 30-32.

فـإذا كـانت لدينا المصفوفة A التالية فإن عملية الفرز التصاعدى تتم بها كالتالى :

بما أن عدد خلايا المصفوفة = ٨ فإن الفراغ = ٤



فى المرحلة الثانية يتم قسمة الفراغ منتصف المصفوفة إلى نصفين ليمثل الفراغ الذى سيستخدم فى المرحلة الثانية .

الفراغ = ٤ ÷ ٢ = ٢

أى مقارنة قيمة الخانة الأ ولى بقيمة الخانة الثالثة، وقيمة الخانة الثانية بقيمة الخانة الرابعة، وهكذا ...

المصفوفة بعد المرحلة الأولى من الفرز

↑1	↓2	3↑	4↓	5↑	6↓	7↑	8↓
45	8	12	15	70	75	20	30

وبعد انتهاء المرحلة الثانية من الفرز (المقارنة) نجد أن المصفوفة ستكون كالتالى :

12	8	45	15	20	30	70	75

وفى المرحلة الثالثة سيكون الفراغ = ٢ ÷ ٢ = ١ أى مقارنة كل قيمة بالقيمة التى تليها ، وعليه سيكون ناتج المقارنة كالتالى :

8	12	15	20	30	45	70	75

والبرنامج التالى يبين خطوات هذا الأسلوب فى الفرز .

شكل (١٤ ــ ٢)

برنامج للحصول على أرقام عشوائية باستخدام دالة RND وفرزها باستخدام أسلوب SHELL

```
10 DIM A(100).
20 GOSUB 50:REM   استدعاء برنامج فرعي للحصول على الا قام العشوائيه
30 GOSUB 120 :REM   استدعاء برنامج فرعي للا جراء عملية الفرز
40 END
50 CLS
60 LOCATE 10,20: INPUT "  ادخل عدد الا رقام المراد فرزها";N
70 FOR I=1 TO N:REM  للحصول على ارقام عشوائيه
80 LET B = RND
90 LET A(I) = INT(B*100)
100 NEXT I
110 RETURN
120 REM  shell  برنامج فرعي للا جراء عملية الفرز باستخدام نظرية
130 LET S$ = TIME$:REM  للحصول على بداية الوقت
140 LET M = 1
150 LET M = M * 2
160 IF M < N THEN 150
170 LET M = INT((M-1) / 2):REM   للا يجاد منتصف المصفوفه
180 IF M = 0 THEN 310
190 LET S = N-M:REM   لتحديد البدايه
200 FOR I = 1 TO S
210 LET J = I
220 LET E = J + M:REM  لتحديد النهايه
```

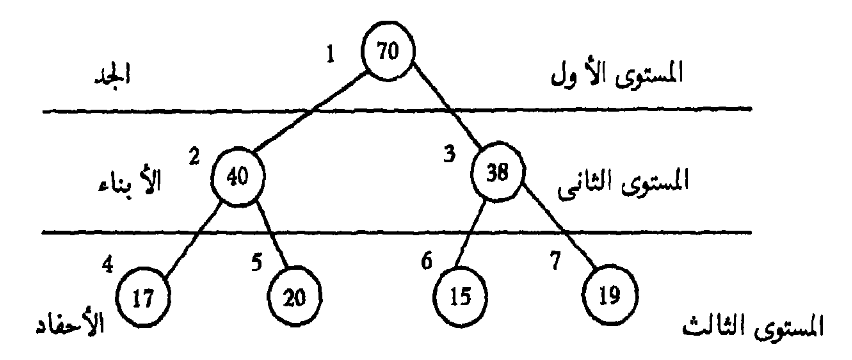
```
230 IF A(E) >= A(J) THEN 290:REM لا جراء المقارنه بين القيم
240 LET T = A(J):REM لم يتحقق الشرط فيتم تغيير اماكن القيم
250 LET A(J) = A(E)
260 LET A(E) = T
270 LET J = J - M:REM طرح عدد ما تم فرزه
280 IF J > 0 THEN 220
290 NEXT I
300 GOTO 170
310 LET F$=TIME$
320 LOCATE 13,20
330 PRINT "باستخدام نظرية shell تم فرز ";N;" من الارقام "
340 PRINT " الوقت من ";S$;" الى ";F$;"
350 RETURN
```

مثال (١٤ ــ ٣) : (الأسلوب الشجرى HEAP) :

إن خطوات الفرز بهذا الأسلوب تشابه عملية رسم شجرة العائلة حيث تبدأ بالجد فالأ بناء فالأحفاد، والمثال التالى يبين كيفية عمل ذلك . فإذا كان لدينا عائلة مكونة من سبعة أفراد وأعمارهم كالتالى :

19 , 15, 20, 17, 38, 40, 70

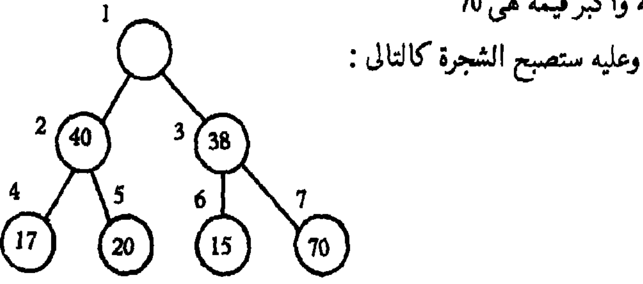
فإن شجرة العائلة ستكون



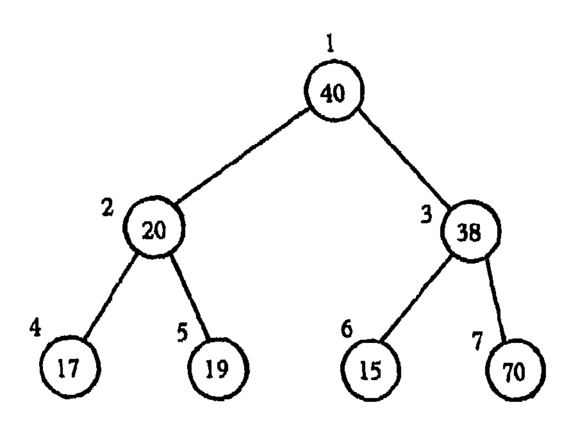
فخطوات الفرز تتم :

١) بإيجاد أكبر قيمة ووضعها فى آخر دائرة، وفى مثالنا هذا آخر دائرة (خلية) هى رقم سبعة وأكبر قيمة هى 70

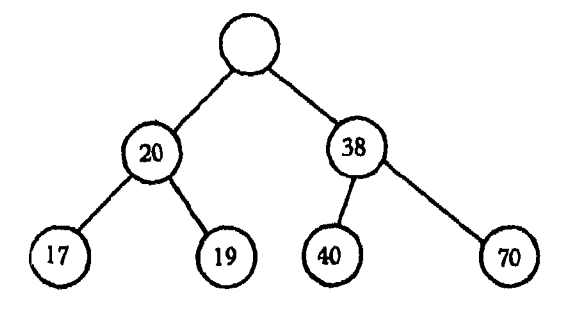
وعليه ستصبح الشجرة كالتالى :



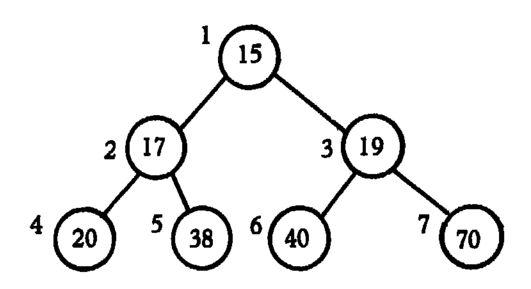
٢) ومن ثم نحتفظ بالقيمة (19) ونبحث عن القيمة الكبرى التالية والتى ستكون من المستوى الثانى (الأ بناء)، وهنا نجد أن القيمة 40 هى التالية فنضعها فى المستوى الأ ول ونبحث عن أحد أتباعها ليحل محلها وهوإما (20 أو 17)، وبما أن القيمة 20 أكبر فننقلها إلى الخانة رقم ٢ وعليه ستكون الشجرة كالتالى :



٣) بعد ذلك نضع القيمة الكبرى (40) فى الخلية التالية التى تم استخدامها، وعليه ستكون الشجرة كالتالى، أى تم إحلال القيمة (40) فى الخلية رقم (6) ونحتفظ بالقيمة السابقة (15) حتى نجد لها الخانة الملائمة، وذلك بتكرار الخطوة الثانية .



٤) بعد تكرار كل من الخطوات ١ ، ٢ ، ٣ ، سنحصل على الشجرة مفروزة كالتالى :



لاحظ أن الشجرة المفروزة بهذا الأسلوب تختلف عن شجرة العائلة بشيئين :

١) أن الخانة الأولى تحتوى على أصغر قيمة .

٢) أنها معكوسة .

والبرنامج التالى يبين كيفية استخدام هذا الأسلوب فى الفرز :

شكل (١٤ ــ ٣)

برنامج للحصول على أرقام عشوائية باستخدام دالة RND وفرزها باستخدام أسلوب HEAP

```
10 DIM A(1000)
20 GOSUB 50:REM استدعاء برنامج للحصول على ارقام عشوائية
30 GOSUB 140 :REM استدعاء برنامج فرعى لا جراء عملية الفرز
40 END
50 REM   RND استدعاء برنامج فرعىللحصول على ارقام عشوائية باستخدام دالة
60 CLS:LOCATE 10,20:PRINT "ادخل عددالا رقام المراد الحصول عليها:"
70 LOCATE 10,20:INPUT N
80 FOR I= 1 TO N
90 LET B= RND : REM لحصول على رقم عشوائى يقع مابين صفر واقل من واحد
100 REM ١٠٠٠ للحصول على رقم صحيح يقع مابين ١ و
110 LET A(I) = INT (B*1000)
120 NEXT I
130 RETURN
140 REM  HEAP برنامج فرعى لا جراء عملية الفرز باستخدام نظرية
150 LET S$=TIME$ :REM لتحديد بداية وقت الفرز
160 FOR I = N TO 2 STEP -1 :REM بدء الفحص من اسفل الى اعلى
170 LET P = 1:REM اسناد القيمة ١ اموشر المصفوفة
180 FOR J = 2 TO I :REM للحصول على المصفوفة من اعلى الى اسفل
190 REM  لا جراء عملية المقارنة بين القيم
200 IF A(J)>A(P) THEN 260
210 REM  لم يتحقق الشرط فستتم عملية تغيير اماكن القيم
220 LET T = A(I)
```

```
230 LET A(I) = A(P)
240 LET A(P) = T
250 GOTO 270
260 LET P = J:REM تغيير مكان المؤشر
270 NEXT J
280 NEXT I
290 REM    للحصول على وقت نهاية الفرز
300 LET F$=TIME$
310 LOCATE 13,20
320 PRINT " باستخدام نظرية heap ثم فرز ";N;"من الارقام "
330 LOCATE 15,20 :PRINT F$;":الى";S$;": الوقت من "
340 RETURN
```

والجدول التالى يبين الفروقات فى الأوقات المستغرقة لفرز كميات من الأعداد المختلفة باستخدام كل من الأساليب الثلاثة :

عدد الأرقام والوقت / الأسلوب	عشرة أرقام		خمسون رقماً		مئة رقم		خمسمائة رقم	
	من	إلى	من	إلى	من	إلى	من	إلى
الفقاعى	00 : 10 : 09	00:10:10	01 :14:40	01:15:04	01:10:47	00:12:24	00:01:09	00:38:07
BUBBLE	ثانية		٢٤ ثانية		دقيقة و٣٧ ثانية		٣٦ دقيقة و٥٧ ثانية	
شل	01:17:14	01:17:15	01:18:28	01:18:35	01:19:25	01:19:41	01:29:24	01:31:38
SHELL	ثانية		٧ ثوانٍ		١٦ ثانية		دقيقتان و١٤ ثانية	
الشجرى	01:22:26	01:22:27	10:22:56	01:23:21	01:23:48	01:52:27	01:20:17	01:48:35
HEAP	ثانية		٢٥ ثانية		دقيقة و٣٩ ثانية		٢٨دقيقة و١٨ثانية	

ملاحظة : لقد تم تشغيل برامج الأساليب الثلاثة على جهاز آى بى إم وسعة ذاكرته الرقمية (256 K) يعتبر أسلوب الفرز (شل SHELL) من أفضل الأساليب الثلاثة بغض النظر عن عدد الأرقام (السجلات) المراد فرزها ، وذلك حسب الزمن المستغرق فى الفرز كما هو مبين فى الجدول السابق .

البحث : فى حالة البحث (Search) عن بيانات سجل معين فى ملف، سجل دراسى مثلاً، يمكن أن يتم بطريقة تتابعية Sequential أى قراءة بيانات السجل الأول، ومقارنة بيانات الحقل المراد بالقيمة المدخلة . وفى حالة المساواة تعرض البيانات، وإلا نستمر فى قراءة بيانات سجل آخر حتى نهاية الملف فإما أن نجد السجل المراد أو لا نجده .

ولو فرضنا أن عدد السجلات الموجودة مائة سجل/وأن بيانات السجل المراد كانت فى السجل تسعة وتسعين، فهذا يعنى أننا سنضطر للقيام بتسع وتسعين عملية مقارنة حتى نصل إلى السجل، ولو كان عدد السجلات أكثر من ذلك فسنضطر للانتظار لوقت أطول حتى نحصل على البيانات المرادة .

مثال (١٤ ــ ٤) : (أسلوب البحث الثنائى) :

وسنتعرض الآن لنوع آخر من طرق البحث والتى تستخدم فقط عندما تكون السجلات المراد البحث فيها مفروزة، و يطلق عليها اسم نظرية (البحث الثنائى Binary Search) بسبب الطريقة المستعملة فى عملية البحث .

فلو كان لدينا مصفوفة مكونة من ثمانية عشر سجلا فعملية البحث باستخدام هذه النظرية تتم كالتالى :

١ ــ رقم السجل الأول = ١

٢ ــ رقم السجل النهائى = ١٨

٣ ــ جمع رقم السجل الأول ورقم السجل الأخير، وقسمة الناتج على ٢ والحصول على الناتج بدون كسر، لنستخدم العبارة التالية :

M = INT (L + H / 2)

حيث إن L = المتغير الذى يحتوى على رقم السجل الأول الذى سيبدأ منه البحث .

و H = المتغير الذى يحتوى على رقم السجل النهائى الذى سينتهى إليه البحث .

و M = المتغير الذى يحتوى على رقم السجل الذى سنقارن فيه القيمة المدخلة بقيمة الحقل المرادف .

٤ ــ مقارنة القيمة المدخلة بقيمة الحقل، فإذا حصلت المساواة يتم إخراج البيانات المرادة .

٥ ــ أما فى حالة عدم المساواة، فيتم الاستفسار عما إذا كانت القيمة المدخلة أكبر من قيمة الحقل التى قورنت به .

٦ ــ ففى حالة كون القيمة أكبر، يعنى أن البحث سيتم فى النصف السفلى من المصفوفة، وعليه ستسند قيمة المتغير M مضافاً إليها واحد إلى المتغير L .

٧ ــ أما إذا كانت القيمة أقل، أى أن البحث سيتم فى النصف العلوى من المصفوفة، فيتم طرح واحد من قيمة المتغير M وإسناد قيمته للمتغير H .

٨ ــ الاستفسار عن قيمة المتغير L إذا كانت أكبر من قيمة المتغير H، يعنى ذلك أنه قد تم الوصول إلى نهاية النصف الذى يجب أن يتم البحث فيه وأنه لم يتم العثور على ما نبحث عنه .

وفى كلتا الحالتين يتم العودة والتنفيذ من الخطوة رقم ٣ .

فإذا كانت لدينا المصفوفة A وعدد خلاياها سبع كالتالى :

45, 32, 27, 22, 20, 15, 10

فإن عملية البحث الثنائى تتم كالتالى : القيمة المراد البحث عنها (20) = B

L = 1, H= 7

M = INT (L + M) / 2) = (1 + 7) / 2 = 4

A (M) = 22

هل B أكبر من (A (M لا

A		B	L	H	M	A (M)
1	10	20	1	7	4	22
2	15					
3	20					
4	22					
5	27					
6	32					
7	45					

هل B = A (M) لا

هل (A (M أكبر من B نعم

إذن البحث سيتم فى النصف العلوى

وعليه فإن: 15 2 3 1 20

هل L = H لا

هل B = A(M) لا

هل (A(M أكبر من B لا

إذن البحث سيتم فى النصف السفلى

وعليه فإن 20 3 3 M + 1 = 3 20

هل L = H لا

هل B = A(M) نعم 20 = 20

وفيما يلى البرنامج الذى يعمل على البحث الثنائى :

شكل (١٤ ــ ٤)

برنامج للبحث عن قيمة فى مصفوفة مفروزة باستخدام أسلوب البحث الثنائى BINARY SEARCH

```
10 REM  اسناد القيم الأوليه للمتغيرات
20 LET L = 1
30 LET H = 7
40 REM  لقراءة البيانات المفروزه و وضعها في مصفوفه
50 FOR I =1 TO 7
60 READ A(I)
70 NEXT I
80 INPUT "  ادخل الرقم المراد البحث عنه ",R
100 IF L > H THEN 230
110 REM  للحصول على منتصف المصفوفه
120 LET M = INT((L + H) / 2)
130 REM  استفسار عما اذا تساوت القيمه المدخله وقيمة الخانه
140 IF A(M) = R THEN 250
150 REM  استفسار اذا كانت القيمه المدخله اكبر من قيمة الخانه
160 IF A(M) < R THEN 200
170 REM  طرح واحد من مكان نهاية المصفوفه
180 LET  H = M - 1
190 GOTO 100
200 REM اضافة  واحد امنتصف المصفوفه و جعله بداية البحث
210 LET L = M + 1
220 GOTO 100
230 PRINT " هذا الرقم  ";R;"  لا  يوجد في المصفوقه"
240 GOTO 270
250 PRINT M;" =  الرقم الذي أدخل  = ";R;" موجود في المصفوفه و مكانه"
260 DATA 10,15,20,22,27,32,45
270 END
```

و يتم تنفيذ البرنامج والحصول على النتائج كالتالى :

القيمة التى تم إدخالها 20 أدخل الرقم المراد البحث عنه

الرقم الذى أدخل = 20 موجود فى المصفوفة ومكانه = 3

تطبيق حكومى

مثال (١٤ ــ ٥) : (الفرز والدمج والبحث) :

الهدف : شعبتان من الدارسين لبرنامج معين، أرادت إدارة القسم أن تعمل على دمج الشعبتين فى مجموعة واحدة، وإيجاد المعدل العام للامتحانين اللذين تقدم إليهما كل دارس والتقدير العام، ومن ثم فرز المجموعة تنازلياً حسب المعدل العام . وإمكانية الاستفسار عن بيانات دارس، أو الحصول على تقرير عن جميع الدارسين، أو الحصول على الدارسين الحاصلين على تقدير معين .

أولاً ــ خطوات الحل :

١ ــ الحصول على عدد دارسى كل من الشعبتين .

٢ ــ قراءة بيانات كل شعبة (الاسم، درجة الامتحان الأول، درجة الامتحان الثانى) . ووضعها فى مصفوفة .

٣ ــ دمج الشعبتين فى مصفوفة جديدة .

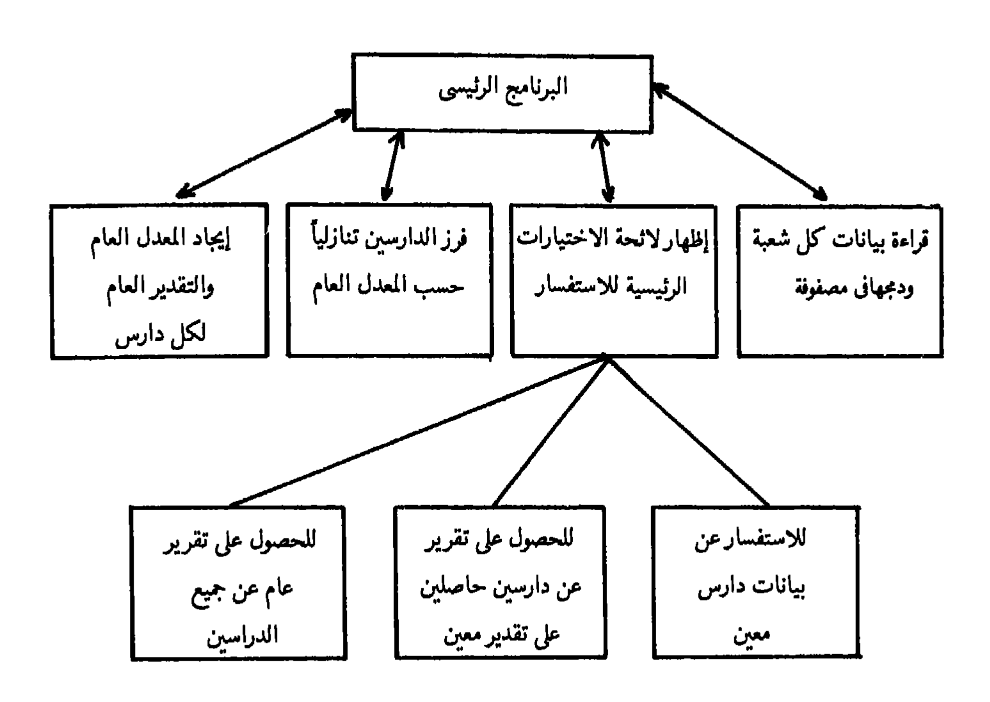
٤ ــ إيجاد المعدل العام والتقدير العام لكل دارس .

٥ ــ فرز الدارسين تنازلياً حسب المعدل العام .

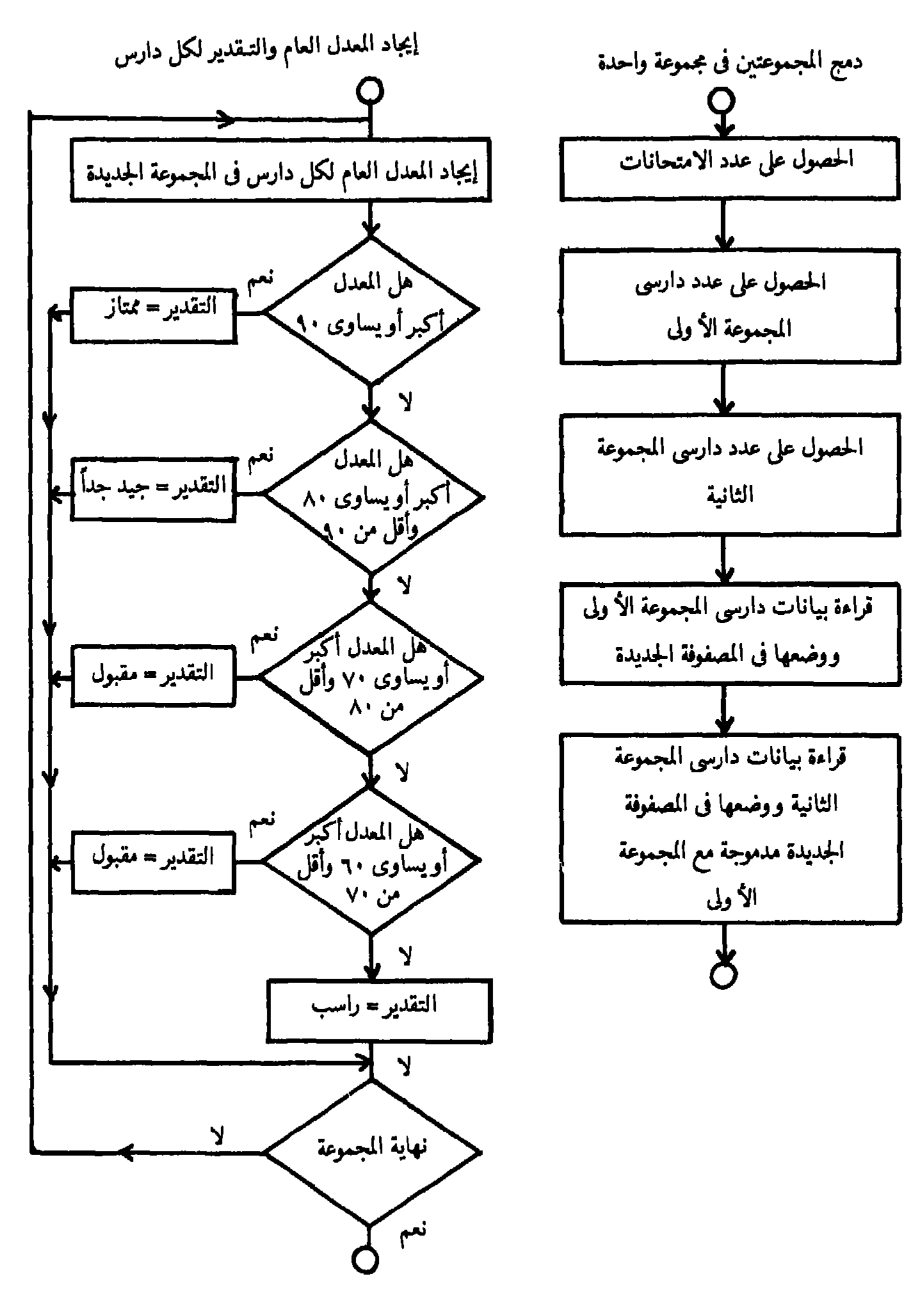
٦ ــ إظهار لائحة الاختيارات الرئيسية .

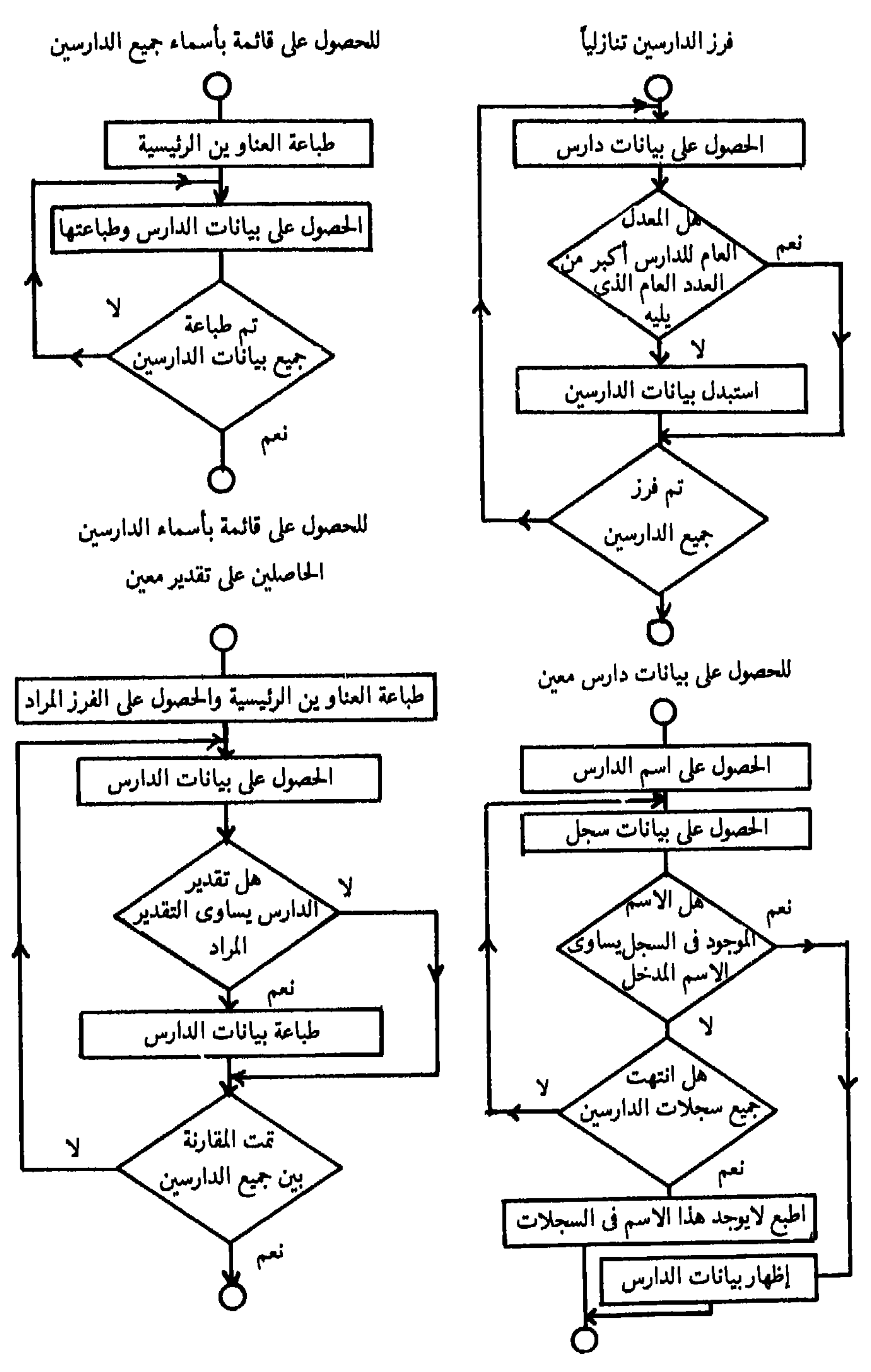
٧ ــ إجراء العمليات اللازمة حسب الاختيار المدخل فى الخطوة السابقة .

ثانياً ــ الهيكل الهرمى للبرنامج :



ثالثاً ــ نمط البرمجة التركيبية :




للحصول على قائمة بأسماء جميع الدارسين
طباعة العناوين الرئيسية
الحصول على بيانات الدارس وطباعتها
تم طباعة جميع بيانات الدارسين
لا
نعم
فرز الدارسين تنازلياً
الحصول على بيانات دارس
هل المعدل العام للدارس أكبر من العدد العام الذى يليه
نعم
لا
استبدل بيانات الدارسين
تم فرز جميع الدارسين
للحصول على قائمة بأسماء الدارسين الحاصلين على تقدير معين
طباعة العناوين الرئيسية والحصول على الفرز المراد
الحصول على بيانات الدارس
هل تقدير الدارس يساوى التقدير المراد
لا
نعم
طباعة بيانات الدارس
تمت المقارنة بين جميع الدارسين
لا
نعم
للحصول على بيانات دارس معين
الحصول على اسم الدارس
الحصول على بيانات سجل
هل الاسم الموجود فى السجل يساوى الاسم المدخل
نعم
لا
هل انتهت جميع سجلات الدارسين
لا
نعم
اطبع لايوجد هذا الاسم فى السجلات
إظهار بيانات الدارس

رابعاً ـ البرنامج كاملاً فى الشكل التالى :

شكل (١٤ ـ ٥)

برنامج لدمج مجموعتين من الدارسين وإيجاد المعدل العام والتقدير لكل دارس، ومن ثم إمكانية الاستفسار

```
10 REM  البرنامج الرئيسي
20 DIM N$(50),N2$(50),N3$(50),G$(50),T1(50,3),T2(50,3),T3(50,3),S(50),A(50)
30 REM  استدعاءبرنامج فرعي لقراءة المجموعتين ودمجهما في مجموعة واحده
40 GOSUB 120
50 REM  استدعاء برنامج فرعي لايجاد المعدل العام والتقدير العام لكل دارس
60 GOSUB 440
70 REM  استدعاء برنامج فرعي لفرز المجموعه حسب المعدل العام تنازليا
80 GOSUB 620
90 REM  استدعاء برنامج فرعي لاجراء العمليات الازمه
100 GOSUB 870
110 END
120 CLS:LOCATE 8,20:PRINT "ادخل عـــــدد الامتحانات "
130     LOCATE 8,20:INPUT T
140     LOCATE 10,20:PRINT "ادخل عدد الدارسين في المجموعه الاولى ";
150     LOCATE 10,20:INPUT N1
160     LOCATE 12,20:PRINT "ادخل عدد الدارسين في المجموعة الثانيه ";
170     LOCATE 12,20:INPUT N2
180 FOR I=1 TO N1
190 READ N1$(I):LET N3$(I)=N1$(I)
200 FOR J=1 TO T
210 READ T1(I,J)
```

```
220 LET T3(I,J)=T1(I,J):NEXT J
230 NEXT I
240 LET I = I - 1
250 REM   للحصول على دارسي المجموعه الثانيه ودرجاتهم ووضعها في مصفوفة B
260 REM   ومن ثم دمجها مع مجموعة A(اي اسنادها لمصفوفة C )
270 FOR K=1 TO N2
280 READ N2$(K):LET N3$(I+K)=N2$(K)
290 REM   للحصول على الدرجات واضافتها للمصفوفه
300 FOR J=1 TO T
310 READ T2(K,J):LET T3(I+K,J)=T2(K,J)
320 NEXT J
330 NEXT K
340 DATA MOHAMMAD,90,70,88
350 DATA ALI,80,90,77
360 DATA SALIH,70,60,89
370 DATA AHMED,70,80,90
380 DATA KHALID,80,90,78
390 DATA SAID,70,60,59
400 DATA MAJED,90,70,88
410 DATA NASIR,80,70,79
420 DATA WLEED,60,50,80
430 DATA OSAMA,60,70,80
440 REM   برنامج فرعي لحساب معدلا ت وتقارير دارسي المجموعتين
450 REM   بعد دمجهم في مصفوفة C
460 REM   للحصول على عدد الدارسين الكلي
470 LET N=N1+N2
480 FOR I=1 TO N
490 REM   لحساب المعدل العام لكل دارس
500 FOR J=1 TO T
```

```
510 LET S(I)=S(I)+T3(I,J)
520 NEXT J
530 LET A(I)=S(I)/T
540 REM                                        لا يجاد التقدير العام لكل دارس
550 IF A(I) >= 90 THEN G$(I)=" ممتـــــــان"
560 IF A(I) < 90 AND A(I) >= 80 THEN G$(I)="جيد جـــــدا"
570 IF A(I) < 80 AND A(I) >= 70 THEN G$(I)="جيـــــــــد"
580 IF A(I) < 70 AND A(I) >= 60 THEN G$(I)=" مقبـــــول"
590 IF A(I) < 60     THEN G$(I) =" راســـــب"
600 NEXT I
610 FOR W=1 TO 10000:NEXT W
620 REM                    برنامج فرعي لفرز الدارسين تنازليا حسب معدلا تهم
630 FOR I=1 TO N-1
640 FOR J=I+1 TO N
650 REM
660 IF A(I) > A(J) THEN 850
670 REM                    لم يتحقق الشرط لذافستتم عملية تغيير الاماكن
680 LET N4$=N3$(I)
690 LET N3$(I)=N3$(J)
700 LET N3$(J)=N4$
710 REM                                        لا ستبدال اماكن الدرجات
720 FOR K=1 TO 3
730 LET R=T3(I,K)
740 LET T3(I,K)=T3(J,K)
750 LET T3(J,K)=R
760 NEXT K
770 REM                                        لا ستبدال اماكن المعدل العام
780 LET R2=A(I)
```

```
790 LET A(I)=A(J)
800 LET A(J)=R2
810 REM                                   لا ستبدال اماكن التقدير العام
820 LET G1$=G$(I)
830 LET G$(I)=G$(J)
840 LET G$(J)=G1$
850 NEXT J
860 NEXT I
870 REM                                   برنامج فرعي لا ظهار شاشة الا ختيار
880 CLS
890 LOCATE 7,10:PRINT "=====================================================
==="
900 LOCATE 8,10:PRINT "                 لا ئحة الا ختيارات الرئيسيه    "
910 LOCATE 9,10:PRINT "                 نوع العمليه
رمـ "
920 LOCATE 10,10:PRINT "----------------------------------------------------  -
--- "
930 LOCATE 11,10:PRINT "          للحصول على قائمه باسماءجميع الدارسين
1 "
940 LOCATE 12,10:PRINT " معين تقدير على الحاصلين الدارسين باسماء قائمة على للحصول
2 "
950 LOCATE 13,10:PRINT "                 للحصول على بيانات دارس معين
3 "
960 LOCATE 14,10:PRINT "                 لا نهـــــــاء العمليــــات
4 "
970 LOCATE 15,10:PRINT "=====================================================
==="
980 LOCATE 16,10:PRINT "          المطلوبه العمليه رمز ادخل"
990 LOCATE 16,10:INPUT O
```

```
1000 REM        للتدقيق والاستفسار عن رمز العمليه المطلوبه
1010 IF O >  0 AND O <  5 THEN 1040
1020 PRINT "        xxxxxxxxxxxxxxxx خطاء في الرمز المدخل xxxxxxxxxxx"
1030 GOTO 980
1040 ON O GOSUB 1070,1170,1330,1560
1050 IF O <  4 THEN 870
1060 RETURN
1070 REM   برنامج فرعي للحصول على قائمةباسماءودرجات وتقاديرجميع الدارسين
1080 CLS
1090 PRINT " التقدير العام   المعدل العام  امتحان 3  امتحان 2  امتحان 1   الاسم"
1100 PRINT "______________  ____________  __________  _________  _________   ______"
1110 FOR I=1 TO N
1120 PRINT USING "\              \    ####.#     ###.#     ###.#     ###.#      \
     \";G$(I),A(I),T3(I,3),T3(I,2),T3(I,1),N3$(I)
1130 PRINT "----------------------------------------------------------------------"
1140 NEXT I
1150 FOR W=1 TO 6000: NEXT W
1160 RETURN
1170 REM   برنامج فرعي للحصول على قائمةباسماء الدارسين الحاصلين على تقدير معين
1180 CLS
1190 LOCATE 10,30:PRINT "        ادخل التقديـــــــر المعـين"
1200 LOCATE 10,30:INPUT E$
1210     LOCATE 2,20:PRINT  " التقـــــــــديـر                     الاســـــــم"
1220     LOCATE 3,20:PRINT  "________________                     ____________"
1230 REM R=    متغير يحوي عددالدارسين الحاصلين على التقدير المطلوب
1240 LET R=0
1250 FOR I=1 TO N
1260 IF G$(I) <> E$ THEN 1290
1270 LET R=R+1
```

```
1280 LOCATE R+3,20:PRINT USING "\                    \                    \
 \ ";G$(I),N3$(I)
1290 NEXT I
1300 LOCATE R+4,20:PRINT "--------------------------------------------------------"
1310 LOCATE R+5,20:PRINT R;"=";E$;" عددالدارسين الحاصلين على تقدير"
1315 FOR I = 1 TO 5000:NEXT I
1320 RETURN
1330 REM      برنامج فرعي للحصول على بيانات دارس معين
1340 CLS
1350 LOCATE 4,30:PRINT ": ادخل اسم الدارس المراد الاستفسار عن بياناته"
1360 LOCATE 4,15:INPUT S$
1370 FOR I=1 TO N
1380 IF N3$(I) <> S$ THEN 1530
1390 LOCATE 6,40:PRINT  ":                    الاســــــــــم"
1400 LOCATE 7,40:PRINT  ":               درجة الامتحان الاول"
1410 LOCATE 8,40:PRINT  ":              درجة الامتحان الثاني"
1420 LOCATE 9,40:PRINT  ":              درجة الامتحان الثالث"
1430 LOCATE 10,40:PRINT ":                     المعدل العام"
1440 LOCATE 11,40:PRINT ":                    التقدير العام"
1450 LOCATE  6,25:PRINT N3$(I)
1460 LOCATE  7,32:PRINT T3(I,1)
1470 LOCATE  8,32:PRINT T3(I,2)
1480 LOCATE  9,32:PRINT T3(I,3)
1490 LOCATE 10,25:PRINT A(I)
1500 LOCATE 11,25:PRINT G$(I)
1510 FOR W=1 TO 6000: NEXT W
1520 GOTO 1550
1530 NEXT I
1540 LOCATE 6,30:PRINT " تأكد من ذلك";S$;" لا يوجد هذا الاسم"
```

```
1550 RETURN
1560 REM    برنامج فرعي لانهاء العمليات
1570 CLS
1580 FOR I=30 TO 50
1590 LOCATE 7,I:PRINT "=";
1600 NEXT I
1610 FOR I=7 TO 14
1620 LOCATE I,30:PRINT ":"
1630 LOCATE I,50:PRINT ":"
1640 NEXT I
1650 FOR I=30 TO 50
1660 LOCATE 14,I:PRINT "=";
1670 NEXT I
1680 LOCATE 11,33:PRINT " مع السلامه "
1690 RETURN
```

وعند تنفيذ البرنامج نتبع الخطوات التالية :

أ ــ تحديد عدد الامتحانات وعدد الدارسين للمجموعتين.

أدخل عدد الامتحانات ? 3

أدخل عدد الدارسين فى المجموعة الأولى ? 4

أدخل عدد الدارسين فى المجموعة الثانية ? 2

ب ــ ظهور لائحة الاختيارات الرئيسية :

رمز	لائحة الاختيارات الرئيسية	نوع العملية
1	للحصول على قائمة بأسماء جميع الدارسين	
2	للحصول على قائمة بأسماء الدارسين الحاصلين على تقدير معين	
3	للحصول على بيانات دارس معين	
4	لإنهاء العمليات	

أدخل رمز العملية المطلوبة ? 1

جـ ــ وعند اختيار الرقم «1» تظهر النتائج كالتالى :

الإسم	امتحان 1	امتحان 2	امتحان 3	المعدل العام	التقدير العام
KHALID	80.0	90.0	78.0	82.7	جيد جدا
MOHAMMAD	90.0	70.0	88.0	82.7	جيد جدا
ALI	80.0	90.0	77.0	82.3	جيد جدا
AHMED	70.0	80.0	90.0	80.0	جيد جدا
SALIH	70.0	60.0	89.0	73.0	جيد
SAID	70.0	60.0	59.0	63.0	مقبول

د ــ وعند اختيار الرقم «3» تظهر النتائج كالتالى :

أدخل اسم الدارس المراد الاستفسار عن بياناته : ALI ؟

الاسم	:	ALI
درجة الامتحان الأول	:	80
درجة الامتحان الثانى	:	90
درجة الامتحان الثالث	:	77
المعدل العام	:	82. 33334
التقدير العام	:	جيدجداً

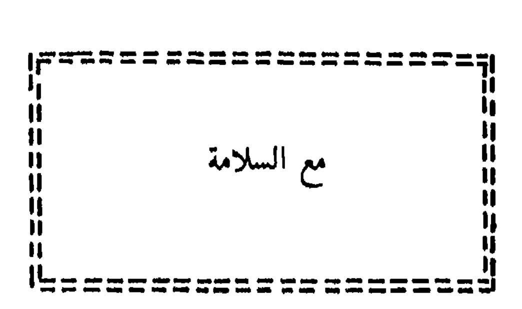
هـ ــ أما الاختيار رقم «2» فيتطلب حالة خاصة، حيث الصعوبة فى استخدام رمز «ــ» لمد الأحرف . ولكن إذا حذفنا رمز المد من التقديرات، يمكن تنفيذ البرنامج كالتالى وذلك بعد اختيار رقم «٢» من الشاشة الرئيسية .

أدخل التقدير المعين جيد جداً؟

فتظهر النتائج كالتالى :

الاســـــم	الـــتـــقـــديـــر
KHALID	جيد جدا
MOHAMMAD	جيد جدا
ALI	جيد جدا
AHMED	جيد جدا
عدد الدارسين الحاصلين على تقدير	جيد جدا = 4

و ـ وللخروج من النظام، يمكن اختيار رقم «4»، فتظهر الشاشة التالية :



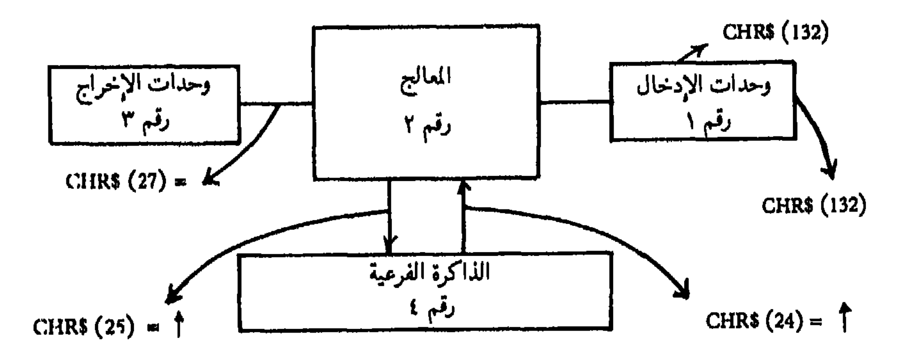
تطبيق تعليمى

مثال (١٤ ـــ ٦) :

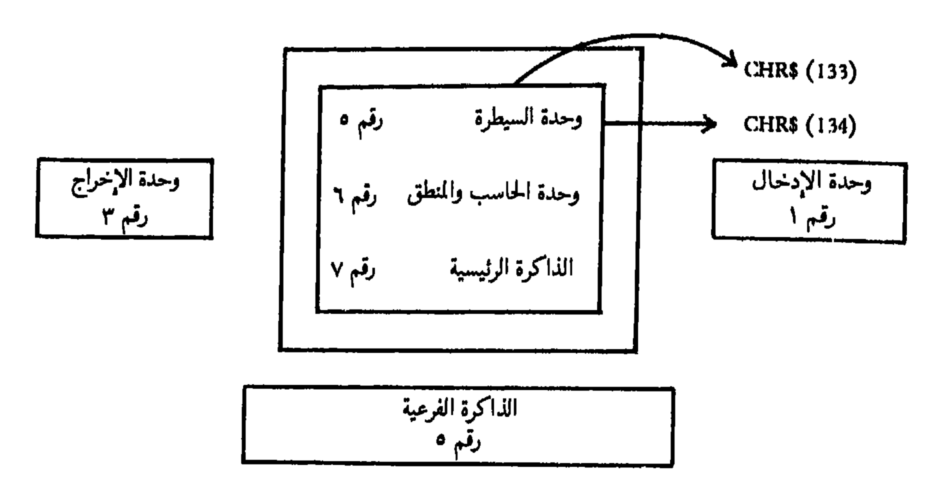
الهدف : إظهار الأجزاء الرئيسية التى يتكون منها الحاسب الآلى ومكونات كل منها ، ومن ثم إظهار كيفية انسياب البيانات والتحكم فى العمليات التى يقوم بها الحاسب .

أولاً ـــ خطوات الحل :

١ ـــ إظهار وحدات الإدخال ، الإخراج ، المعالج والذاكرة الفرعية . كما هو مبين فى الشكل التالى :

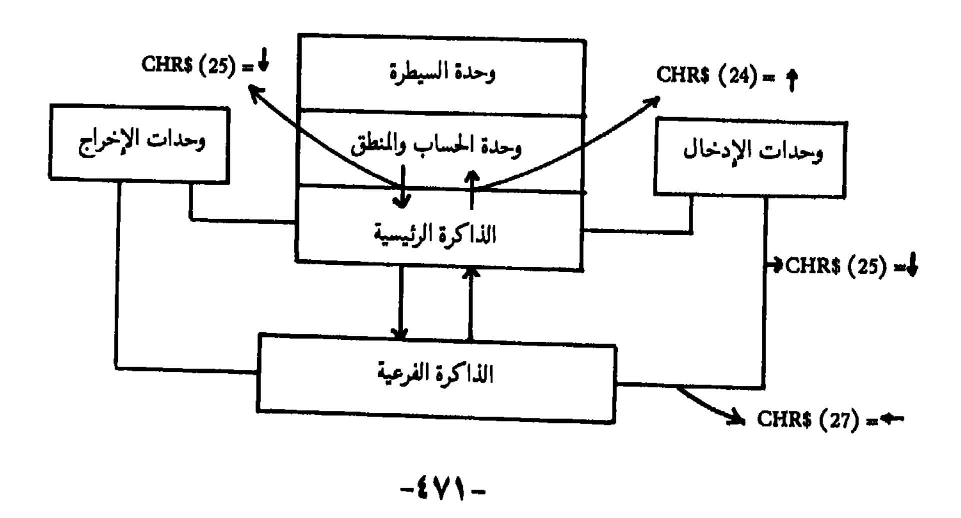


٢ ــ إظهار أجزاء المعالج كما هو مبين بالشكل التالى :

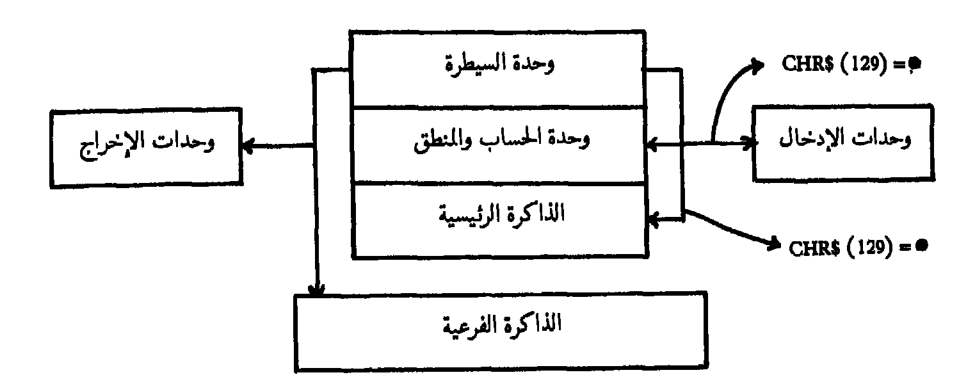


وسنشير إلى أرقام المربعات فى البرنامج لتسهيل تتبع التنفيذ . كما أشير إلى نوع الرمز الذى يستخدم للخطوط كلها بواسطة الدالة CHR$.

٣ ــ إظهار كيفية انسياب البيانات بين الأجزاء المختلفة :

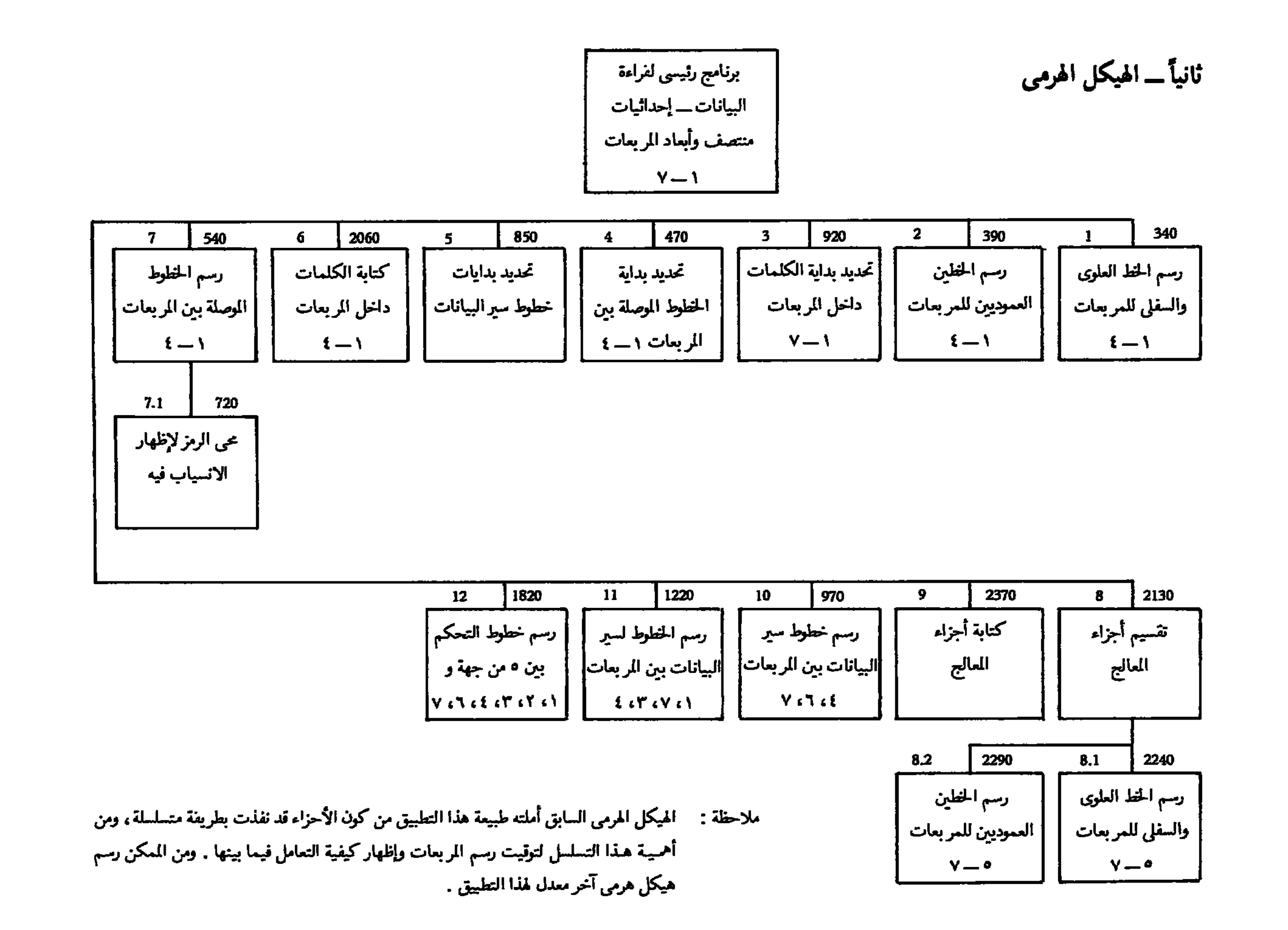


٤ ـ إظهار كيفية القيام بالتحكم :



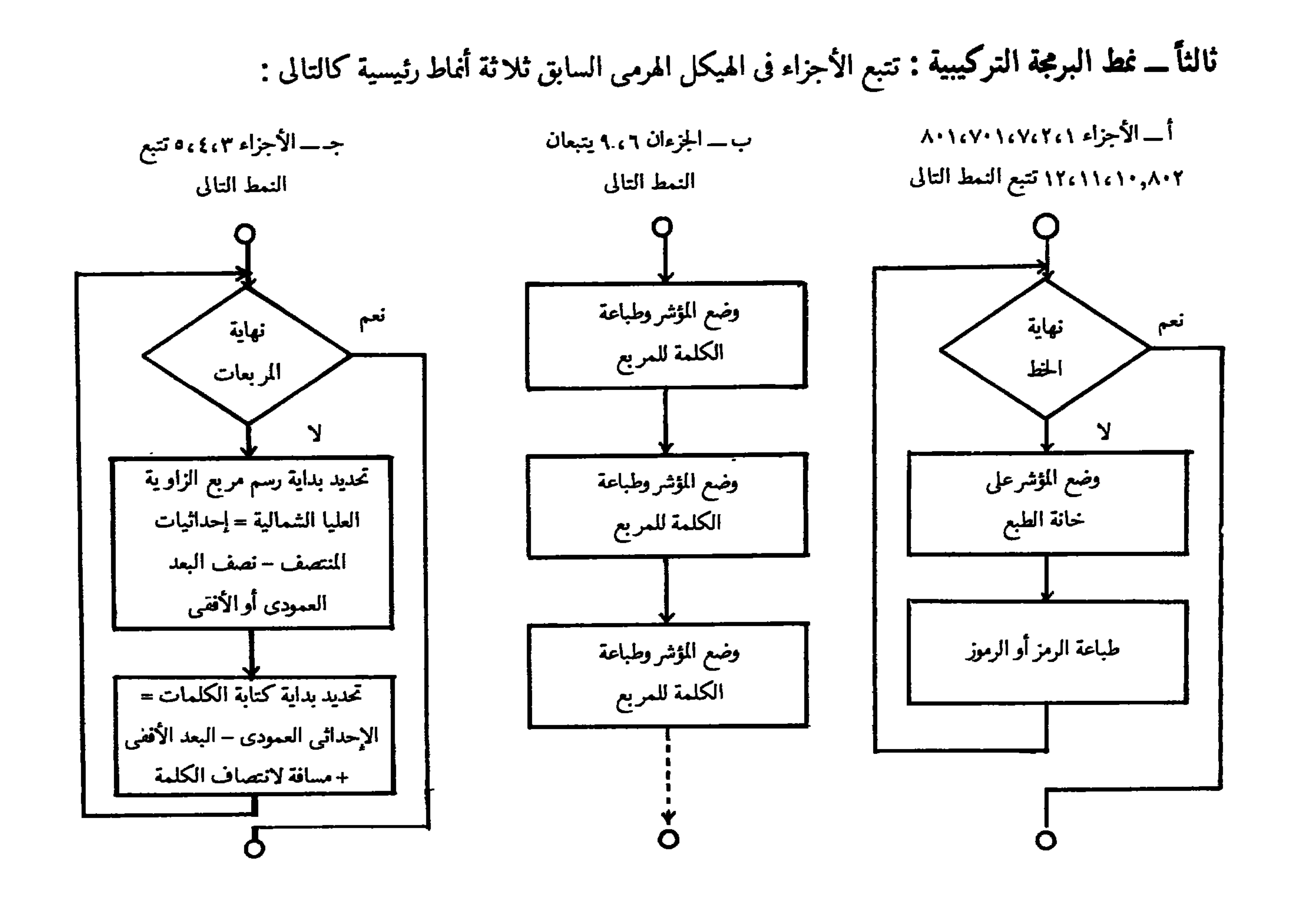
حيث سيتم استخدام الدالة CHR$ لرسم بعض الرموز، وتعليمة COLOR للتحكم فى ألوان الشاشة . وقد تم شرحهما فى الفصل السادس عشر .

ثانياً — الهيكل الهرمى



ملاحظة : الهيكل الهرمى السابق أملته طبيعة هذا التطبيق من كون الأجزاء قد نفذت بطريقة متسلسلة، ومن أهمية هذا التسلسل لتوقيت رسم المربعات وإظهار كيفية التعامل فيما بينها . ومن الممكن رسم هيكل هرمى آخر معدل لهذا التطبيق .

ثالثاً — نمط البرمجة التركيبية : تتبع الأجزاء فى الهيكل الهرمى السابق ثلاثة أنماط رئيسية كالتالى :



رابعاً ــ البرنامج كاملاً فى الشكل التالى :

شكل (١٤ ــ ٦)

برنامج لإظهار الأجزاء الرئيسية للحاسب، وانسياب البيانات وكيفية التحكم

```
10 KEY OFF:REM  للتمكن من استخدام اخر سطر في الشاشه
20 CLS
30 COLOR 1,0
40 COLOR 12
50 FOR I = 1 TO 7
60 READ CN(I,1),CN(I,2)
70 READ DM(I,1),DM(I,2)
80 NEXT I
90 FOR I = 1 TO 4
100 R1 = CN(I,1)- INT(DM(I,1)/2)
110 R2 = CN(I,2)- INT(DM(I,2)/2)
120 LOCATE R1,R2
130 GOSUB 340:REM  رسم الخط العلوي للمربعات
140 GOSUB 390:REM  رسم الخطوط العموديه
150 LOCATE R1+DM(I,1),R2
160 GOSUB 340
170 FOR A=1 TO 800:NEXT A
180 NEXT I
190 GOSUB 920 REM  تحديد بداية كتابة الكلمات داخل المربعات الاساسية
200 GOSUB 470 REM  تحديد بدايات الخطوط الموصلة بين المربعات
210 GOSUB 850 REM  تحديد بدايات الخطوط الموصلة بين المربعات لسير البيانات
```

```
220 GOSUB 2060 REM كتابة الكلمات داخل المربعات الاساسية
230 GOSUB 540 REM رسم الخطوط الموصلة
240 LOCATE W(2,1),W(2,2)+8:PRINT "          "
250 GOSUB 2130 REM تقسيم اجزاء المعالج
260 FOR P = 1 TO 500:NEXT P
270 GOSUB 2370:REM لكتابة اجزاء المعالج
280 GOSUB 970 REM رسم الخطوط الموصلة بين الذاكره الرئيسية والفرعيةوالحساب
290 GOSUB 1220:REM رسم الخطوط الموصله للبيانات بين المربعات
300 GOSUB 1820:REM رسم خطوط التحكم بين وحدة التحكم و المربعات
310 FOR Z =1 TO 5000:NEXT Z
320 LOCATE 25,1
330 END
340 REM رسم الخط العلوي للمربع
350 FOR M = 1 TO DM(I,2)
360 PRINT CHR$(132);
370 NEXT M
380 RETURN:END
390 REM رسم الخطين العمودين للمربع
400 FOR K = 1 TO DM(I,1)
410 LOCATE R1+K, R2
420 PRINT CHR$(132)
430 LOCATE R1+K,R2+DM(I,2)-1
440 PRINT CHR$(132)
450 NEXT K
460 RETURN:END
470 REM تحديد بدايت الخطوط الموصلة بين المربعات
480 FOR T =1 TO 7
490 IF T=1 OR T=2 OR T=5 THEN B(T,1)=CN(T,1):B(T,2)=CN(T,2)-INT(DM(T,2)/2)
500 IF T=3 OR T=6 THEN 520
```

```
510 IF T=4 OR T=7 THEN B(T,2)=CN(T,2):B(T,1)=CN(T,1)-INT(DM(T,1)/2)
520 NEXT T
530 RETURN:END
540 REM رسم الخطوط الموصلة بين المربعات
550 COLOR 1,2
560 FOR Q = 1 TO 3
570 FOR S=1 TO 13
580 FOR T=1 TO 4
590 IF T=1 THEN LOCATE B(T,1),B(T,2)-S:PRINT CHR$(27)
600 IF T=2 THEN LOCATE B(T,1),B(T,2)-S:PRINT CHR$(27)
610 IF T=3 THEN 660
620 IF S > 4 THEN 660
630 IF T=4 THEN LOCATE B(T,1)-S,B(T,2)+4:PRINT CHR$(24):LOCATE B(T,1)-5+S,B(T,2)
-4:PRINT CHR$(25)
640 FOR E = 1 TO 15:NEXT E
650 SOUND 40,3
660 NEXT T
670 SOUND 32767 ,1
680 NEXT S
690 GOSUB 720
700 NEXT Q
710 RETURN:END
720 REM محي الخطوط الموصلة بين المربعات
730 COLOR 1,2
740 IF Q = 9 THEN 840
750 FOR C=1 TO 13
760 FOR V=1 TO 4
770 IF V=1 THEN LOCATE B(V,1),B(V,2)-C:PRINT " "
780 IF V=2 THEN LOCATE B(V,1),B(V,2)-C:PRINT " "
```

```
790 IF V=3 THEN 820
800 IF C > 4 THEN 820
810 IF V=4 THEN LOCATE B(V,1)-C,B(V,2)+4:PRINT " ":LOCATE B(V,1)-5+C,B(V,2)-4:PR
INT " "
820 NEXT V
830 NEXT C
840 RETURN:END
850 REM حساب بدايات الخطوط التفصيلية
860 FOR T =1 TO 7
870 IF T=2 OR T=5 OR T=6 THEN 900
880 IF T=1 OR T=3 THEN Z1(T,1)=CN(T,1)+INT(DM(T,1)/2):Z1(T,2)=CN(T,2)
890 IF T=4 OR T=7 THEN Z1(T,1)=CN(T,1):Z1(T,2)=CN(T,2)-INT(DM(T,2)/2)
900 NEXT T
910 RETURN:END
920 REM تحديد بدايات كتابة الكلمات داخل المربعات
930 FOR T =1 TO 7
940 W(T,1)=CN(T,1):W(T,2)=CN(T,2)-INT(DM(T,2)/2)+1
950 NEXT T
960 RETURN:END
970 REM ==========================================
980 COLOR ,2
990 FOR Q=1 TO 3
1000 FOR L=1 TO 5
1010 FOR Y=1 TO 4
1020 LOCATE B(4,1)-Y,B(4,2)-4:PRINT CHR$(24):LOCATE B(4,1)-5+Y,B(4,2)+4:PRINT CH
R$(25)
1030 NEXT Y
1040 REM
1050 FOR Y =1 TO 4
```

```
1060 LOCATE B(4,1)-Y,B(4,2)-4:PRINT " ":LOCATE B(4,1)-5+Y,B(4,2)+4:PRINT " "
1070 FOR H=1 TO 40:NEXT H
1080 NEXT Y
1090 NEXT L
1100 FOR L=1 TO 5
1110 FOR Y=1 TO 3
1120 LOCATE B(7,1)-Y,B(7,2)-4:PRINT CHR$(24):LOCATE B(7,1)-4+Y,B(7,2)+4:PRINT CH
R$(25)
1130 FOR H = 1 TO 30 :NEXT H
1140 NEXT Y
1150 REM
1160 FOR Y=1 TO 3
1170 LOCATE B(7,1)-Y,B(7,2)-4:PRINT " ":LOCATE B(7,1)-4+Y,B(7,2)+4:PRINT " "
1180 NEXT Y
1190 NEXT L
1200 NEXT Q
1210 RETURN :END
1220 REM
1230 FOR G =1 TO 7
1240 LOCATE B(1,1),B(1,2)-G
1250 PRINT CHR$(27)
1260 NEXT G
1270 FOR H = 1 TO 5
1280 LOCATE B(1,1)-1+H,B(1,2)-7
1290 PRINT CHR$(25)
1300 NEXT H
1310 FOR P = 1 TO 6
1320 LOCATE B(1,1)+4,B(1,2)-7-P
1330 PRINT CHR$(27)
```

```
1340 NEXT P
1350 GOSUB 850
1360 FOR G=1 TO 6
1370 LOCATE Z1(7,1),Z1(7,2)-G-2:PRINT CHR$(27)
1380 NEXT G
1390 FOR G=1 TO 5
1400 LOCATE Z1(7,1)+1-G,Z1(7,2)-8:PRINT CHR$(24):NEXT G
1410 FOR G=1 TO 7
1420 LOCATE Z1(7,1)-4,Z1(7,2)-8-G:PRINT CHR$(27):NEXT G
1430 FOR G=1 TO 10
1440 LOCATE Z1(1,1)+G,Z1(1,2):PRINT CHR$(25):NEXT G
1450 FOR G=1 TO 7
1460 LOCATE Z1(1,1)+10,Z1(1,2)-G:PRINT CHR$(27):NEXT G
1470 FOR G=1 TO 6
1480 LOCATE Z1(4,1),Z1(4,2)-G:PRINT CHR$(27):NEXT G
1490 FOR G=1 TO 10
1500 LOCATE Z1(4,1)-G+1,Z1(4,2)-7:PRINT CHR$(24):NEXT G
1510 FOR P=1 TO 2000:NEXT P
1520 REM
1530 COLOR 0,2
1540 FOR G =1 TO 7
1550 LOCATE B(1,1),B(1,2)-G
1560 PRINT " "
1570 NEXT G
1580 FOR H = 1 TO 5
1590 LOCATE B(1,1)-1+H,B(1,2)-7
1600 PRINT " "
1610 NEXT H
1620 FOR P = 1 TO 6
```

```
1630 LOCATE B(1,1)+4,B(1,2)-7-P
1640 PRINT " "
1650 NEXT P
1660 FOR G=1 TO 6
1670 LOCATE Z1(7,1),Z1(7,2)-G-2:PRINT " "
1680 NEXT G
1690 FOR G=1 TO 5
1700 LOCATE Z1(7,1)+1-G,Z1(7,2)-8:PRINT " ":NEXT G
1710 FOR G=1 TO 7
1720 LOCATE Z1(7,1)-4,Z1(7,2)-8-G:PRINT " ":NEXT G
1730 FOR G=1 TO 10
1740 LOCATE Z1(1,1)+G,Z1(1,2):PRINT " ":NEXT G
1750 FOR G=1 TO 7
1760 LOCATE Z1(1,1)+10,Z1(1,2)-G:PRINT " ":NEXT G
1770 FOR G=1 TO 6
1780 LOCATE Z1(4,1),Z1(4,2)-G:PRINT " ":NEXT G
1790 FOR G=1 TO 10
1800 LOCATE Z1(4,1)-G+1,Z1(4,2)-7:PRINT " ":NEXT G
1810 RETURN:END
1820 REM =============================================
1830 COLOR 16,2
1840 FOR G=1 TO 7 STEP 2
1850 LOCATE B(5,1),B(5,2)+23+G:PRINT CHR$(129):FOR F=1 TO 50:NEXT F:NEXT G
1860 FOR G=1 TO 5
1870 LOCATE B(5,1)+G-1,B(5,2)+30:PRINT CHR$(129):FOR F=1 TO 50:NEXT F:NEXT G
1880 FOR G=2 TO 6 STEP 2
1890 LOCATE B(5,1)+4,B(5,2)+30+G:PRINT CHR$(129):FOR F=1 TO 50: NEXT F:NEXT G
1900 FOR G=1 TO 7 STEP 2
1910 LOCATE B(5,1),B(5,2)-G-2:PRINT CHR$(129):FOR F=1 TO 50:NEXT F:NEXT G
```

```
1920 FOR G=1 TO 5
1930 LOCATE B(5,1)+G-1,B(5,2)-9:PRINT CHR$(129):FOR F=1 TO 50:NEXT F:NEXT G
1940 FOR G=2 TO 6 STEP 2
1950 LOCATE B(5,1)+4,B(5,2)-9-G:PRINT CHR$(129):FOR F=1 TO 50: NEXT F:NEXT G
1960 FOR G=1 TO 15
1970 LOCATE B(5,1)+G-1,B(5,2)-9:PRINT CHR$(129):FOR F=1 TO 50:NEXT F:NEXT G
1980 FOR G=1 TO 7 STEP 2
1990 LOCATE B(5,1)+2,B(5,2)+31-G:PRINT CHR$(129):NEXT G
2000 FOR G=1 TO 8
2010 LOCATE B(5,1)+G-1,B(5,2)+30:PRINT CHR$(129):FOR F=1 TO 50:NEXT F:NEXT G
2020 FOR G=1 TO 7 STEP 2
2030 LOCATE B(5,1)+8,B(5,2)+31-G:PRINT CHR$(129):NEXT G
2040 COLOR 1,2
2050 RETURN:END
2060 FOR T = 1 TO 4
2070 IF T = 1 THEN LOCATE W(T,1),W(T,2)+1:COLOR 30,0:PRINT " مد خلات ":FOR T1=1 T
O 200:NEXT T1
2080 IF T = 2 THEN LOCATE W(T,1),W(T,2)+8:COLOR 14,0: PRINT " المعا لج ":FOR T1
= 1 TO 200:NEXT T1
2090 IF T = 3 THEN LOCATE W(T,1),W(T,2)+1:COLOR 30,0:PRINT " مخرجات ":FOR T1 =1
TO 200:NEXT T1
2100 IF T = 4 THEN LOCATE W(T,1),W(T,2)+17:PRINT " ذا كر ة فر عية ":FOR T1=1 TO
200:NEXT T1
2110 NEXT T
2120 RETURN:END
2130 REM   تقسيم اجزاء المعالج
2140 FOR I = 5 TO 7
2150 R1 = CN(I,1)- INT(DM(I,1)/2)
2160 R2 = CN(I,2)- INT(DM(I,2)/2)
```

```
2170 LOCATE R1,R2
2180 GOSUB 2240
2190 GOSUB 2290
2200 LOCATE R1+DM(I,1),R2
2210 GOSUB 2240
2220 FOR A=1 TO 800:NEXT A
2230 NEXT I
2240 REM رسم الخط العلوي للمربع
2250 FOR M = 1 TO DM(I,2)
2260 PRINT CHR$(133);
2270 NEXT M
2280 RETURN:END
2290 REM رسم الخطين العموديين للمربع
2300 FOR K = 1 TO DM(I,1)
2310 LOCATE R1+K, R2
2320 PRINT CHR$(134)
2330 LOCATE R1+K,R2+DM(I,2)-1
2340 PRINT CHR$(134)
2350 NEXT K
2360 RETURN:END
2370 REM كتابة اجزاء المعالج
2380 FOR T = 5 TO 7
2390 IF T=6 THEN LOCATE W(T,1),W(T,2):COLOR 14,0:PRINT " وحدة الحساب والمنطق ":F
OR T1=1 TO 200:NEXT T1
2400 IF T=5 THEN LOCATE W(T,1),W(T,2)+3:COLOR 14,0: PRINT " وحدة التحكم ":FOR T1
=1 TO 200:NEXT T1
2410 IF T=7 THEN LOCATE W(T,1),W(T,2)+1:COLOR 14,0: PRINT " الذاكرة الرئيسية ":F
OR T1=1 TO 200:NEXT T1
```

```
2420 NEXT T
2430 RETURN:END
2440 DATA 8,73,6,12    (١) احداثيات منتصف ابعاد مربع المدخلا ت
2450 DATA 8,41,12,26   (٢) احداثيات منتصف وابعاد مربع المعالج
2460 DATA 8,09,6,12    (٣) احداثيات منتصف وابعاد مربع المخرجات
2470 DATA 21,41,4,50   (٤) احداثيات منتصف وابعاد مربع الذاكره الفرعيه
2480 DATA 04,41,2,22   (٥) احداثيات منتصف وابعاد مربع وحدة التحكم
2490 DATA 06,41,2,22   (٦) احداثيات منتصف وابعاد مربع وحدة الحساب والمنطق
2500 DATA 12,41,2,22   (٧) احداثيات منتصف وابعاد مربع وحدة الذاكره الرئيسيه
```

وعند تنفيذ البرنامج ، ستظهر الرسومات متتالية على الشاشة شبيهة بتلك المشروحة فى مقدمة المثال .

تمارين

١ ــ اعمل على طباعة وتنفيذ البرامج الموجودة فى أشكال الفصل، وإجراء التعديلات عليها إن وجدت، وتنفيذها حسب تسلسل هذه التعديلات .

٢ ــ يتسلم محل لبيع التمور طلبيات بواسطة ثلاثة طرق : الهاتف، البريد، وشخصى . يبيع المحل ثلاثة أصناف من التمور . وفى نهاية كل يوم يقوم المدير بتحضير قائمة لكمية المبيعات كالتالى :

أنواع التمور:

أسلوب الطلبية	الأول	الثانى	الثالث
الهاتف	xxx	xxx	xxx
البريد	xxx	xxx	xxx
شخصى	xxx	xxx	xxx

وإذا علم أن مصفوفة الأسعار للكيلو كالتالى :

أنواع التمور:

أسلوب الطلبية	الأول	الثانى	الثالث
الهاتف	١٥	٢٠	٣٠
البريد	١٢	١٧	٢٥
شخصى	١٠	١٥	٢٠

وإذا كان هناك خصم على المجموع الكلى لقيمة المبيعات حسب الجدول التالى :

أنواع التمور :

أسلوب الطلبية	الأول	الثانى	الثالث
الهاتف	٦	٧	٩
البريد	٣	٥	٧
شخصى	٢	٣	٤

فالمطلوب :

١ ـ طور برنامجاً لحساب مجموع كمية المبيعات حسب نوع التمور وحسب أسلوب الطلبية والمجموع الكلى .

٢ ـ طور برنامجاً لحساب قيمة المبيعات لكل نوع من التمور وأسلوب الطلبية والمجموع الكلى .

٣ ـ طور برنامجاً لحساب صافى الربح لكل نوع وأسلوب المجموع الكلى .

٤ ـ اعمل على إنشاء ملف ليحتوى على مجاميع بيع كل صنف من أصناف التمور الثلاثة خلال الشهر .

٥ ـ اعمل على إنشاء ملف ليحتوى على مجاميع الربح الصافى لكل صنف من أصناف التمور خلال الشهر .

٦ ـ طور برنامجاً لحساب مجموع المبيعات والأرباح لكل صنف شهرياً .

الفصل الخامس عشر

السلسلات : أساليب التعامل معها وتطبيقاتها

مقدمة عن السلسلات STRINGS

تعرضنا فى الفصل السابق لكيفية إجراء المعالجات الحسابية على البيانات الرقمية فقط ، وكيفية إجراء معالجات الإدخال والفرز والدمج والبحث والإخراج للبيانات بنوعيها : الرقمى ، وغير الرقمى . وتشير «السلسلات» إلى أرقام وأحرف أدخلت إلى الحاسب على شكل بيانات غير خاضعة للعمليات الحسابية ، فالبيانات «RIYADH 11141 S.A.» تعتبر سلسلة مع كونها محتوية على أرقام.

وهناك عمليات خاصة يمكن إجراؤها على السلسلات ولها تعليمات ذات وظائف محددة من بينها :

١ ــ اختيار عدد معين من الأحرف بدءاً من يسار السلسلة .

٢ ــ اختيار عدد معين من الأحرف بدءاً من يمين السلسلة .

٣ ــ اختيار عدد معين من الأحرف من منتصف السلسلة .

٤ ــ معرفة عدد الأحرف فى سلسلة معينة .

بالإضافة إلى ذلك ، يمكن دمج سلسلتين بواسطة علامة + .

وسنشرح فى الجزء التالى التعليمات التى تؤدى هذه الوظائف ، بالإضافة إلى تعليمات أخرى ذات علاقة بالتعامل مع السلسلات .

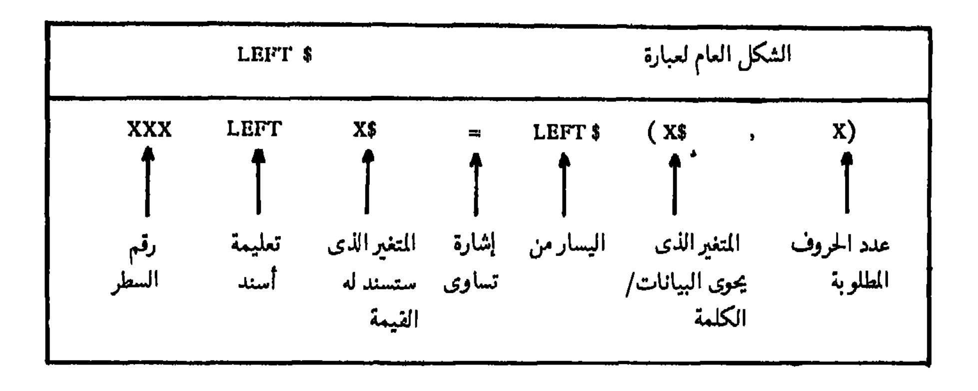
تعليمات السلسلات

الأيسر : إذا أردنا إدخال اسم شخص ما على ثلاث مراحل ، كالاسم الأول واسم الأب واسم العائلة ، ومن ثم طباعة الاسم فى سطر واحد بالصيغة التالية : الاسم الأول ، الحرف الأول من اسم الأب ، اسم العائلة ، فإن ذلك

يتطلب الحصول على الانسم الأول ، اسم الأب ، واسم العائلة ، كما هو موضح فى العبارات التالية :

```
10  REM     قراءة الاسم الأول ، اسم الأب ، واسم العائلة
20  READ    NI$, N3$
30  DATA    «ALI», «AHMED», «ALHUSINEE»
```

أما طباعة اسم العائلة ، الحرف الأول من اسم الأب ، والاسم الأول فعند طلب الحصول على الحرف الأول فقط من اسم الأب . ولتحقيق ذلك ، يمكننا استخدام الدالة LEFT $ كما هومبين فى الشكل التالى :



وعليه سنسند الحرف الأول من اسم الأب للمتغير $ F 2 كما يلى :

40 LET F 2 $ = LEFT $ (N2 $,1)

اسم المتغير الذى يحتوى على اسم الأب

عدد الحروف المرادة من أقصى اليسار (وهنا الحرف الأول)

50 PRINT N3$; F2 $; N1$

60 END

ولو قمنا بتنفيذ هذه العبارات سنحصل على المخرجات كالتالى :

ALHUSINEE A. ALI

وفى مثالنا هذا يمكننا طباعة الأسماء دون استخدام الفاصلة أو الفاصلة المنقوطة أو غيرها من دالات التحكم فى الطباعة، وذلك باستخدام إشارة الزائد الجبرية (+) كالتالى :

50 PRINT N3$ + « » + F2$ + « . » + N1$

ولو قمنا بتنفيذ العبارات بعد التعديل على عبارة رقم ٥٠ لحصلنا على المخرجات بالشكل التالى :

ALHUSINEE A. ALI

و يطلق على هذه الطريقة اسم التوصيل CONCATINATION أى وضع هذه القيم أو قيم المتغيرات بلصق بعضها مع بعض، ولهذا استخدمنا مابين كل متغير وآخر وقبل إشارة الزائد علامتى التنصيص « » ووضعنا مسافتين بين العلامتين حتى يتم فصل الطباعة بعضها عن بعض .

الأيمن : إذا أردنا الحصول على حرف أو أكثر من يمين الكلمة نستخدم الدالة RIGHT$ كما هو مبين فى الشكل التالى :

الشكل العام لعبارة الدالة						RIGHT$
X)	(X$,	RIGHT$	=	X$	LET	XXX
↑	↑	↑	↑	↑	↑	↑
عدد الحروف المرادة من اليمين للخلف	المتغير الذى يحتوى على القيمة	من يمين القيمة	إشارة يساوى	المتغير الذى ستسند له القيمة	تعليمة أسند	رقم السطر

فمثلاً إذا كان المتغير A $ يحوى القيمة «ABDULLAH»

فإن :

ــ RIGHT$ (A$,2) يساوى AH أى أول حرفين من يمين قيمة A$

ــ LEFT$ (A$, 2) يساوى AB أى أول حرفين من يسار قيمة A$

الوسط : كما أمكننا الحصول على أجزاء من كلمة من يسارها أو يمينها يمكننا الحصول على أجزاء من وسطها ؛ وذلك باستخدام الدالة MID$ كما هو مبين فى الشكل التالى :

MID$				الشكل العام لعبارة الدالة			
XXX	LET	X$	=	MID$	(X$,	X ،	X)
↑	↑	↑	↑	↑	↑	↑	↑
رقم السطر	تعليمة أسند	اسم المتغير الذى ستسند له القيمة	إشارة يساوى	من منتصف داخل المتغير	اسم المتغير الذى يحتوى على القيمة	رقم الحرف المراد بدء الإسناد من عنده	عدد الحروف المرادة

فمثلاً : إذا كانت قيمة المتغير A$ تساوى «GOODMORNING» فإن :

ــ MID$ (A$, 5, 7) يساوى MORNING أى الحروف السبعة من بداية العمود الخامس .

ــ LEFT$ (A$, 4) يساوى GOOD أى الحروف الأربعة من اليسار .

ــ RIGHT$ (A$, 7) يساوى MORNING أى الحروف السبعة من أقصى اليمين وللخلف .

ــ MID$ (A$, 1, 4) يساوى GOOD أى الحروف الأربعة من بداية العمود الأول .

لاحظ أنه إلى حد كبير يوجد تشابه ما بين الوظائف التى تؤديها كل من الدالات MID$ و LEFT$ و RIGHT$

فمثلاً إذا كانت قيمة المتغير A$ تساوى

10 LET A$ = «GOODAFTERNOON MORNING EVE»

فإن ١) 20 PRINT LEFT$ (A$, 4) + « » + MID$ (A$, 5, 9)

(GOOD) + مسافة + AFTERNOON

سيطبع GOOD AFTERNOON

٢) PRINT LEFT$ (A$, 4) + « » + MID$ (A$, 14, 7)

GOOD + مسافة + MORNING

سيطبع GOOD MORNING

٣) 40 PRINT MID$ (A$, 1, 4) + « » + RIGHT$ (A$, 3) +MID$ (A$, 17, 4)

GOOD + مسافة + EVE + NING

سيطبع GOOD EVENING

الطول : LEN وإذا أردنا الحصول على عدد الحروف التى تتكون منها كلمة معينة أو اسم معين فيمكننا عمل ذلك باستخدام الدالة LEN كما هومبين فى الشكل التالى :

الشكل العام لعبارة			LEN		
(X$)	LEN	=	X	LET	XXX
اسم المتغير المراد إيجاد طوله (عدد الحروف التى يتكون منها)	دالة الطول	إشارة يساوى	اسم المتغير الذى سيسند له طول المتغير	تعليمة إسناد	رقم السطر

فمثلاً إذا كانت لدينا البيانات التالية (خمسة أسماء) وأردنا إيجاد طول كل اسم (عدد الأحرف التى يتكون منها)، نستخدم العبارات التالية :

```
10 FOR I = 1 TO S
20 READ N$
30 LET A = LEN (N$)
40 PRINT A, «تساوى» ;N$ «عدد الحروف التى يتكون منها الاسم»
50 NEXT I
60 DATA «ALI », «AHMED», «HASAN», «SALIM», «TAMEEM»
70 END
```

ولو قمنا بتنفيذ هذه العبارات فسنحصل على المخرجات التالية :

عدد الحروف التى يتكون منها الاسم	ALI	تساوى	3
عدد الحروف التى يتكون منها الاسم	AHMED	تساوى	5
عدد الحروف التى يتكون منها الاسم	HASAN	تساوى	5
عدد الحروف التى يتكون منها الاسم	SALIM	تساوى	5
عدد الحروف التى يتكون منها الاسم	TAMEEM	تساوى	6

تحويل البيانات من رقمية إلى غير رقمية وبالعكس : فيما مضى تعرضنا لمعالجة البيانات غير الرقمية (خاصة الحروف)، كالحصول على أجزاء منها وتوصيلها بعضها ببعض وما إلى ذلك، وإلى معالجة البيانات الرقمية كأرقام فقط لإجراء العمليات الحسابية عليها ومعالجتها كبيانات غير رقمية فقط فى حالة وجودها بين علامتى التنصيص « » .

وسنتعرض الآن لكيفية تحويل هذه البيانات إلى أصلها أى (بيانات رقمية) وذلك باستخدام الدالة VAL كما هومبين بالشكل التالى :

الشكل العام لعبارة			VAL		
(X$)	VAL	=	X	LET	XXX
المتغير الذى يحتوى على البيانات	دالة القيمة الرقمية	إشارة تساوى	المتغير الذى ستسند له القيمة الرقمية	تعليمة أسند	رقم السطر

فمثلاً إذا كانت قيمة المتغير A$ تساوى «8» و المتغير B$ تساوى «7» لا يمكننا إجراء أى عملية حسابية عليها ــ لأنهما يحو يان قيمتين تعتبران غير عدديتين ؛ لوجود هاتين القيمتين بين علامتى التنصيص . وللحصول على القيمة العددية لكل متغير نستخدم العبارات التالية :

```
10 LET A = VAL (A$)
20 LET B = VAL (B$)
```

و بعد تنفيذ العبارتين نجد أن المتغيرين العدديين A و B قد أسندت لهما القيمتان 8 و 7 بالتوالى . و بإمكاننا بعد ذلك إجراء أى عملية حسابية عليهما كالجمع أو الطرح ... وما إلى غير ذلك .

فمثلاً

```
30 PRINT A + B
```

سيكون ناتج الطباعة 15

أما إذا كانت لدينا بيانات عددية وأردنا تحويلها والتعامل معها كبيانات غير عددية، فنستخدم الدالة STR$ كما هو مبين فى الشكل التالى :

الشكل العام لعبارة الدالة			STR$		
(X)	STR$	=	X$	LET	XXX
↑	↑	↑	↑	↑	↑
المتغير أو القيمة المراد تحويلها	دالة التحويل من قيمة عددية إلى غير عددية	إشارة يساوى	المتغير الذى ستسند له القيمة بعد تحويلها	تعليمة أسند	رقم السطر

فمثلاً لو حصلنا على تاريخ اليوم بالصيغة التالية : اليوم (خانتان)، الشهر (خانتان)، والسنة (أربع خانات) ـ 19871015 ـ وأردنا إخراجه بالصيغة التالية، اليوم/ الشهر/ السنة، نستخدم العبارات التالية :

```
10  LET A#= 19871015
20  LET A$ = STR$ (A #)
30  PRINT LEFT$ (A$,5)+ «/» + MID$ (A$,6,2)+«/» + RIGHT$ (A$,2)
```

وفى حالة تحويل قيمة عددية إلى قيمة غير عددية يتم إضافة خانة إلى يسار الرقم للدلالة على إشارته (سالب/ موجب)، فإذا كان الرقم موجباً فلن تظهر الإشارة. بالإضافة إلى ذلك؛ فقد تم استخدام نوع جديد من المتغيرات (A#)، حيث يدل الرمز (#) بعد اسم المتغير على زيادة القدرة التخزينية للأرقام الصحيحة .

وهناك رموز أخرى تؤثر فى تحديد القيم المخزنة فى المتغيرات ويمكن الرجوع إلى ذلك فى أدلة استخدام اللغة .

وكذلك فإن عدد خانات البيانات العددية المحولة إلى قيم غير عددية يكون أكبر من عدد الخانات الفعلية بواحدة .

و بعد تنفيذ هذه العبارة سنحصل على المخرجات كالتالى :

1987/10/15

أى تمت قراءة التاريخ كقيمة عددية وتم تحويلها إلى قيمة غير عددية، ومن ثم تقسيمها إلى ثلاثة أجزاء وإخراجها بعد القيام بعملية التوصيل .

تطوير برامج باستخدام تعليمات السلسلات

مثال (١٥ ـ ١) :

الهدف : مدير محل معين لبيع قطع ملابس رجالية يريد معرفة عدد القطع المتوفرة عنده من الأصناف التى يبيعها، مع العلم بأن هذه القطع مصنفة حسب التالى :

	الخاصة	الرمز
الصنف :	(خانة واحدة)	
	ـ بدلة (ثلاث قطع)	١
	ـ بدلة (قطعتان)	٢
	ـ بدلة ثلاث قطع سبورت	٣
	ـ بدلة قطعتان سبورت	٤
الحجم :	(ثلاث خانات)	حسب المقاس 36,37,38,39,... 48
	ـ القياس (خانتان)	
	ـ نوعه (خانة واحدة)	
	طويل	L
	عادى	R
	قصير	S

اللون : (خانتان)

ـــ اللون الأساسى أخضر		G
أسود		B
أبيض		W
ـــ طبيعة اللون ـــ غامق		D
فاتح		L

فيصبح الرمز كالتالى :

الخانتان الخامسة والسادسة — الخانات الثانية والثالثة والرابعة — الخانة الأولى

اللون		الحجم		الصنف
الفرعى	الأساسى	النوع	القياس	
D	G	L	3 : 7	1
L	B	R	3 : 8	2
	W	S	:	3
			4 : 8	4

أولاً ـــ خطوات الحل :

١ ـــ الحصول على الاستفسار المراد .

٢ ـــ معرفة رمز البدلة بالحصول على قيمة الخانة الأولى من الاستفسار .

٣ ـــ معرفة الحجم المراد بالحصول على قيمتى الخانتين (٢ ، ٣) من الاستفسار .

٤ ـــ معرفة نوع الحجم بالحصول على قيمة الخانة (٤) .

٥ ـــ معرفة اللون المراد بالحصول على قيمة الخانة (٥) .

٦ ـــ معرفة طبيعة اللون بالحصول على قيمة الخانة (٦) .

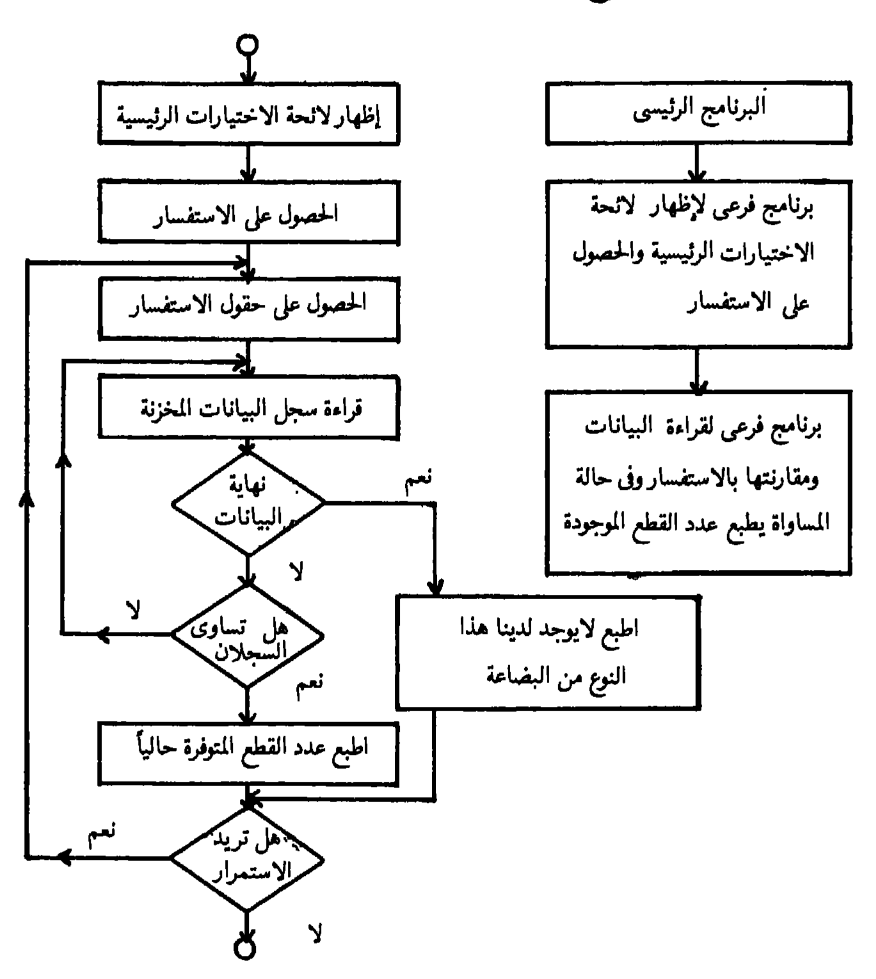
٧ ـــ قراءة سجلات البيانات .

٨ ــ مقارنة قيم الخانات المدخلة بقيم خانات السجل (البيانات) وفى حالة المساواة نحصل على قيم الخانات (٧، ٨، ٩) من سجل البيانات لتساوى الكمية المتوفرة من القطع .

٩ ــ فى حالة وصول نهاية البيانات وعدم حصول مساواة مع الاستفسار، يعنى ذلك عدم توفر القطعة فى المخزن .

ثانياً ــ الهيكل الهرمى للبرنامج :

ثالثاً ــ نمط البرمجة التركيبية :



رابعاً ــ البرنامج فى الشكل التالى :

شكل (١٥ ــ ١)

برنامج للاستفسار عن عدد القطع الموجودة من البدل الرجالية

```
10 REM   البرنامج الرئيسي
20 GOSUB 40
30 END
40 REM   برنامج فرعي لطباعة مواصفات البدله
50 CLS
60 LOCATE 3,20:PRINT  "=========================================="
70 LOCATE 4,20:PRINT  ";          لا ئحة الاختيار الرئيسيه           ;"
80 LOCATE 5,20:PRINT ";-----------------------------------------;"
90 LOCATE 6,20:PRINT ";     الرمز              الصنف              ;"
100 LOCATE 7,20:PRINT ";    ______     ____________________       ;"
110 LOCATE 8,20:PRINT ";      1             بدله ثلا ث قطع        ;"
120 LOCATE 9,20:PRINT ";      2              بدله قطعتين          ;"
130 LOCATE 10,20:PRINT";      3        بدله ثلا ث قطع سبورت       ;"
140 LOCATE 11,20:PRINT";      4         بدله قطعتين سبورت         ;"
150 LOCATE 12,20:PRINT";    99-00         القياس (رقمين )          ;"
160 LOCATE 13,20:PRINT";      L        نوع القياس -طويل           ;"
170 LOCATE 14,20:PRINT";      R                -عادي              ;"
180 LOCATE 15,20:PRINT";      S                -قصير              ;"
190 LOCATE 16,20:PRINT";      B      اللون الا ساسي -أسود         ;"
200 LOCATE 17,20:PRINT";      W                -أبيض              ;"
210 LOCATE 18,20:PRINT";      D      طبيعة اللون - غامق           ;"
220 LOCATE 19,20:PRINT";      L                -فاتح              ;"
230 LOCATE 20,20:PRINT"=========================================="
```

```
240 LOCATE 21,20:PRINT " : أدخل مواصفات البدله المراد الاستفسار عنها"
250 LOCATE 21,12:INPUT A$
260 GOSUB 380:REM
270 CLS:REM  برنامج فرعي للايجاد عدد البدل المتوفره
280 LOCATE 10,20:PRINT "  لا = 2  أو  نعم  = 1  هل تريد الاستمرار أدخل  "
290 LOCATE 10,18:INPUT A
300 IF A = 1 THEN 340
310 IF A = 2 THEN 350
320 LOCATE 20,20:PRINT "xxxxx خطأ في الادخال xxxxxx"
330 GOTO 280
340 RESTORE : GOTO 50
350 REM
360 RETURN
370 REM
380 REM  لقراءة السجل
390 READ R$
400 REM  استفسار عن نهاية البيانات
410 IF R$ = "          " THEN 640
420 REM   للحصول على حقول الاستفسار
430 LET A1$ = LEFT$(A$,1) :REM  نوع البدله
440 LET A2$ = MID$(A$,2,2) :REM  المقاس
450 LET A3$ = MID$(A$,4,1) :REM  نوع المقاس
460 LET A4$ = MID$(A$,5,1) :REM  اللون الاساسي
470 LET A5$ = MID$(A$,6,1) :REM  طبيعة اللون
480 REM  للحصول على سجل البيانات و مقارنتها بحقول الاستفسار
490 LET R1$ = LEFT$(R$,1)
500 LET R2$ = MID$(R$,2,2)
510 LET R3$ = MID$(R$,4,1)
520 LET R4$ = MID$(R$,5,1)
```

```
530 LET R5$ = MID$(R$,6,1)
540 LET R6$ = RIGHT$(R$,3)
550 REM  للا ستفسار اذا تساوت بيانات السجل مع بيانات الا ستفسار
560 IF A1$=R1$AND A2$=R2$ AND A3$=R3$ AND A4$=R4$ AND A5$=R5$ THEN 580
570 READ R$: GOTO 410:REM  قراءة سجل آخر و العوده للا ستفسار
580 CLS: LOCATE 10,30
590 PRINT R6$;" = المتوفرة القطع عدد  "
600 FOR W=1 TO 3000: NEXT W
610 REM
620 FOR I = 1 TO 2000: NEXT I
630 GOTO 660
640 CLS: LOCATE 10,30:PRINT "  لا يوجد هذا الصنف
650 FOR W=1 TO 3000: NEXT W
660 RETURN
670 DATA "140LBL072"
680 DATA "238RBD090"
690 DATA "336SWL101"
700 DATA "444SBL104"
710 DATA "240RWD024"
720 DATA "350LWD073"
730 DATA "138LWL065"
740 DATA "242RBD045"
750 DATA "         "
```

وفى حالة تنفيذ البرنامج السابق سنحصل على التالى :

أ ــ الشاشة الرئيسية التى تبين مواصفات البدلة ورموزها وإدخال المواصفات التالية الموجودة فى البيانات : 240 RWD .

لائحة الاختيار الرئيسية

الصنف	الرمز
بدلة ثلاث قطع	1
بدلة قطعتان	2
بدلة ثلاث قطع سبورت	3
بدلة قطعتان سبورت	4
القياس (رقمان)	99-00
نوع القياس ــ طويل	L
ــ عادى	R
ــ قصير	S
اللون الأساسى ــ أسود	B
ــ أبيض	W
طبيعة اللون ــ غامق	D
ــ فاتح	L

أدخل مواصفات البدلة المراد الاستفسار عنها : 240RWD ؟

ب ــ الإجابة على الاستفسار بالنتائج التالية :

عدد القطع المتوفرة = 024

جـ ــ للاستمرار فى الاستفسار تظهر الشاشة التالية :

هل تريد الاستمرار أدخل 1 = نعم أو 2 = لا 1 ؟

د ــ الاستفسار عن مواصفات غير متوفرة، أدخل مثلاً 244RWL فتظهر الشاشة التالية :

لايوجد هذا الصنف

مثال (١٥ ـ ٢) :

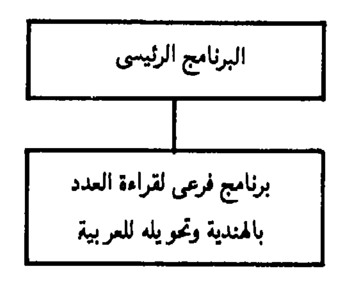
لا تحتوى لغة بيسك على القدرة على إجراء العمليات الحسابية على الأرقام الهندية (١ و ٢ و ٣ .... ٩) لكن لها القدرة على إجراء مثل هذه العمليات على الأرقام العربية (0, 1, 2, ...9) . لذا فى حالة وجود بيانات تحتوى على (أرقام هندية) لغرض إجراء عمليات حسابية عليها ، لابد لنا من التعامل معها وكأنها قيم غير عددية ، ومن ثم إيجاد المرادف لكل قيمة من الأرقام العربية ، وأخيراً إجراء العمليات الحسابية على القيمة المرادفة وتحويل الناتج إلى ما يرادفه من القيم فى الأرقام الهندية .

الهدف : تحويل الأرقام الهندية إلى الأرقام العربية .

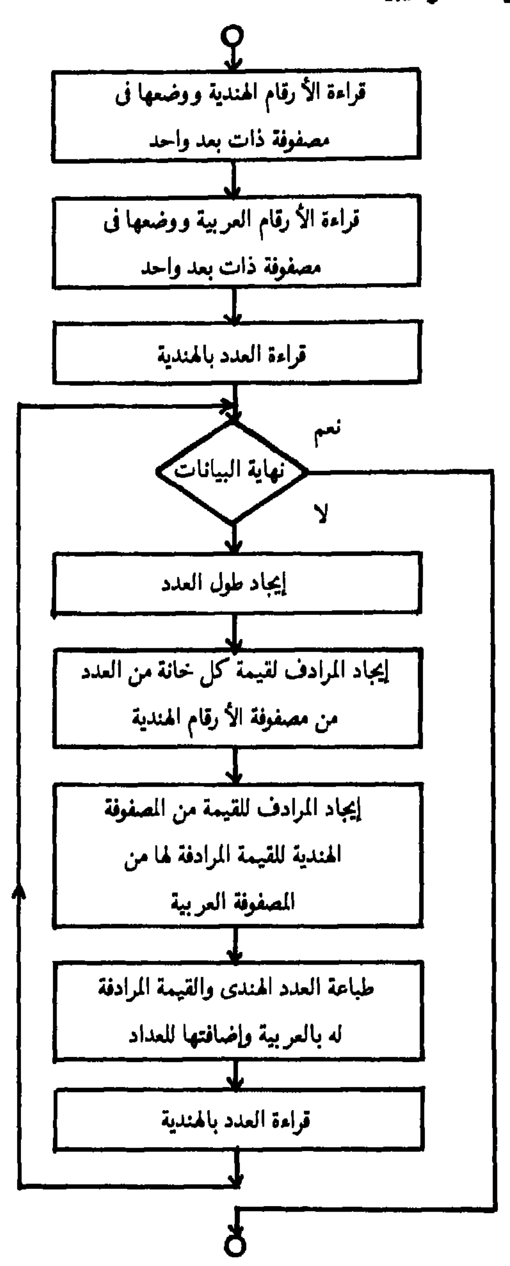
أولاً ـ خطوات الحل :

١ ـ الحصول على قيمة العدد .
٢ ـ إيجاد عدد خاناته (طوله) .
٣ ـ بعدد الخانات نوجد المرادف لقيمة كل خانة من الأرقام العربية .
٤ ـ توصيل الأرقام العربية ببعضها ونوجد قيمتها .
٥ ـ إخراج القيمة للعدد بالعربية والهندية .

ثانياً ـ الهيكل الهرمى للبرنامج :



ثالثاً ــ نمط البرمجة التركيبية :



رابعاً ـــ البرنامج فى الشكل التالى :

شكل (١٥ ـــ ٢)

برنامج لتحويل مجموعة من الأعداد بالعربية إلى ما يرادفها بالهندية

```
10 REM  استدعاء برنامج فرعي
20 GOSUB 50
30 END
40 REM  برنامج فرعي لتحويل ارقام عربيه الى هنديه
50 DIM A$(20),B$(20),C$(20),D(20)
60 FOR I=1 TO 10
70 REM  قراءة البيانات المراد تحويلها
80 READ D(I):R(I) = D(I): NEXT I
90 DATA 34,27,974,22,13,15,67,99,140,358
100 FOR I=1 TO 10
110 REM  قراءة ارقام هنديه ووضعها في مصفوفه
120 READ A$(I):NEXT I
130 DATA ٠,١,٢,٣,٤,٥,٦,٧,٨,٩
140 FOR I =1 TO 10
150 IF D(I) < 100 THEN 220
160 REM  للحصول على قيمة خانة المئات
170 LET T = INT(D(I) / 100)
180 PRINT A$(T+1);
190 REM  للحصول على قيمة خانة العشرات
200 LET D(I) = D(I) - (T * 100)
210 REM  للحصول على قيمة خانة آحاد
220 LET T = INT (D(I) / 10)
```

```
230 PRINT A$(T+1);
240 LET D(I) = D(I) - (T * 10)
250 T = D(I)
260 REM  طباعة البيانات بالعربيه و الهنديه
270 PRINT A$(T+1),R(I)
280 NEXT I
290 RETURN
```

وعند تنفيذ البرنامج، سنحصل على النتائج التالية :

```
34      ٣٤
27      ٢٧
974     ٩٧٤
22      ٢٢
13      ١٣
15      ١٥
67      ٦٧
99      ٩٩
140     ١٤٠
358     ٣٥٨
        OK
```

مثال (١٥ ــ ٣) :

التشفير CRYPTOGRAPHY هو إحدى الطرق للمحافظة على سرية البيانات والمعلومات للحيلولة دون التعرف على محتويات النص فى حالة وقوعها بين أيدى غير المختصين، وتستخدم هذه الطريقة فى المراسلات الحكومية وغيرها من سياسات الشركات، حيث تستخدم شفرة (CIPHER) خاصة لتحويل النص الأصلى (PLAINTEXT) إلى نص الشفرة (CIPHERTEXT) .

و يتم عمل ذلك بإحدى الطرق التالية :

أ) طريقة الأعمدة : حيث يتم اختيار أبعاد المصفوفة المراد وضع النص بها، ومن ثم ترتيب الأعمدة المراد قراءة النص منها، ومن ثم طباعتها .

ب) طريقة الاستبدال : حيث يتم اختيار تسلسل ترتيب الحرف (فى سلسلة الحروف الأ بجدية اللا تينية) الذى سيحل مكان الحرف الحالى من النص .

فمثلاً إذا كان لدينا النص الأصلى التالى NATIONAL TRANSPORTATION وأردنا كتابته بنص الشفرة بطريقة الأعمدة ــ فيتم ذلك حسب الخطوات التالية :

١ ــ تحديد نهاية النص بوضع إشارة ($)

فيكون النص $ NATIONAL TRANSPORTATION

٢ ــ يتم تحديد أبعاد المصفوفة (الأعمدة والأسطر) المراد وضع النص الأصلى بها . بحيث يكون عدد خلايا المصفوفة مساو ياً لعدد حروف النص أو أكثر .

فمثلاً لو كانت الأ بعاد (4, 6) أى ستة أعمدة وأربعة سطور يصبح جدول النص كالتالى :

	1	2	3	4	5	6
1	N	A	T	I	O	N
2	A	L	T		R	A
3	N	S	P	O	R	T
4	A	T	I	O	N	$

٣ ــ يتم اختيار ترتيب الأعمدة لطباعة النص، فمثلاً لو كان الترتيب المراد هو عمود (2, 1, 4, 6, 3, 5) يصبح النص كالتالى :

ALST	NANA	ITOO	NAT$	TPI	ORRN
2	1	4	6	3	5

أما إذا أردنا كتابة النص نفسه بطريقة الاستبدال فنتبع الخطوات التالية :

١ ــ حذف المسافات (الفراغ) ما بين الكلمات ووضع مؤشر نهاية النص ($) وعليه يصبح النص GOODMORNING $.

٢ ــ يتم اختيار تسلسل الحرف فمثلاً لو تم اختيار التسلسل = 3

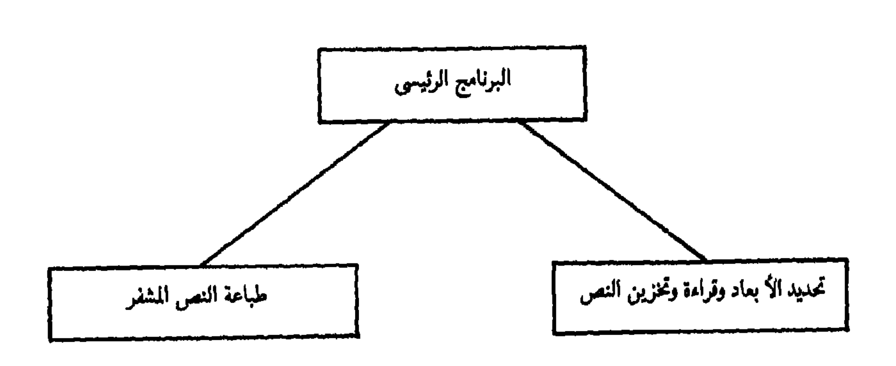
أى يتم استبدال كل حرف بالحرف الثالث الذى يليه من تسلسل الحروف الأبجدية اللاتينية، فمثلاً لو كان الحرف من النص يساوى A يتم استبداله بالحرف D لأن ترتيب D يساوى 3 ما بعد الحرف A وهكذا . وفى حالة الوصول إلى نهاية الحروف الأبجدية اللاتينية Z يتم العودة من بدايتها، فمثلاً لو كان الحرف من النص يساوى X يتم استبداله بالحرف A .

والبرنامج التالى سيعمل على تطبيق الطريقة الأولى :

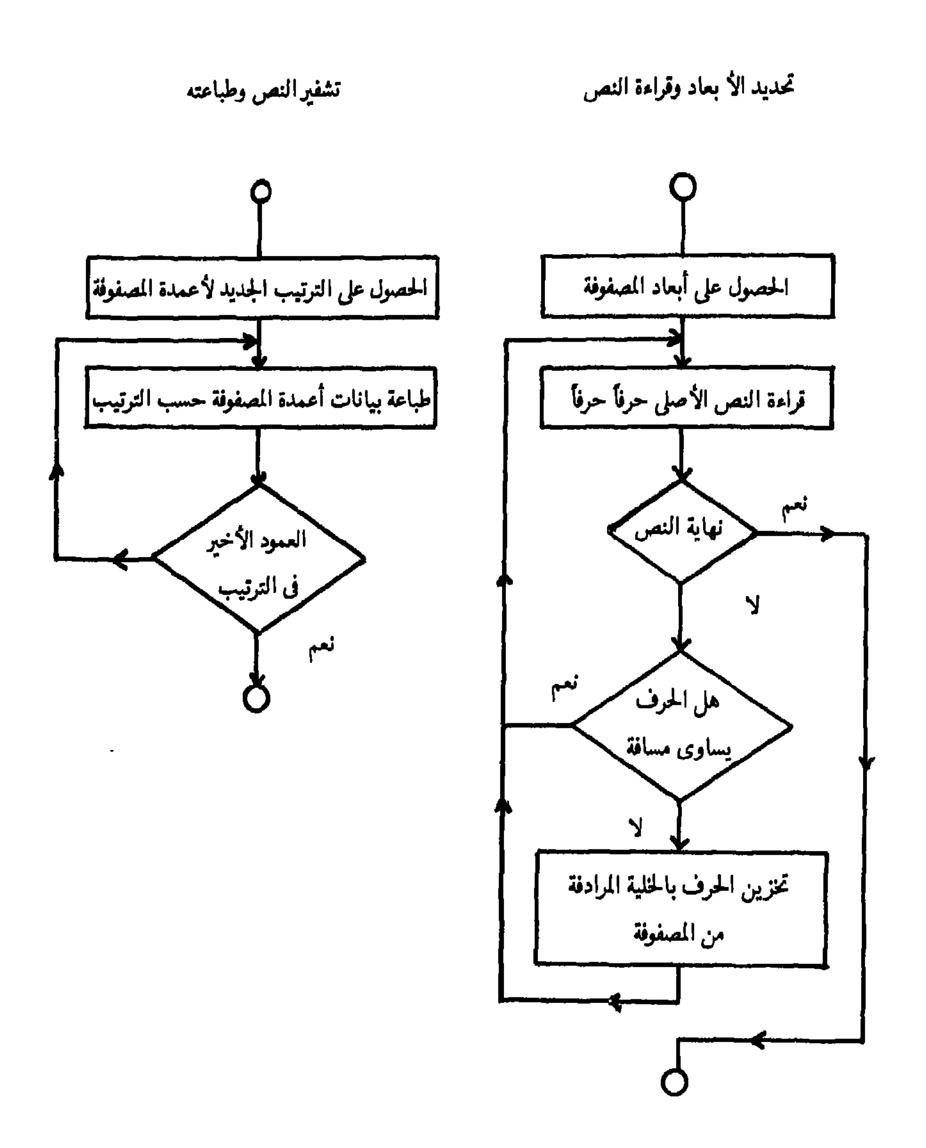
الهدف : تشفير البيانات قبل تخزينها وإعادة فك الشفرة وطباعة النص الأصلى .

أولاً ــ الخطوات : كما ذكرت سابقاً.

ثانياً ــ الهيكل الهرمى :



ثالثاً ـــ نمط البرمجة التركيبية :



رابعاً ــ البرنامج فى الشكل التالى :

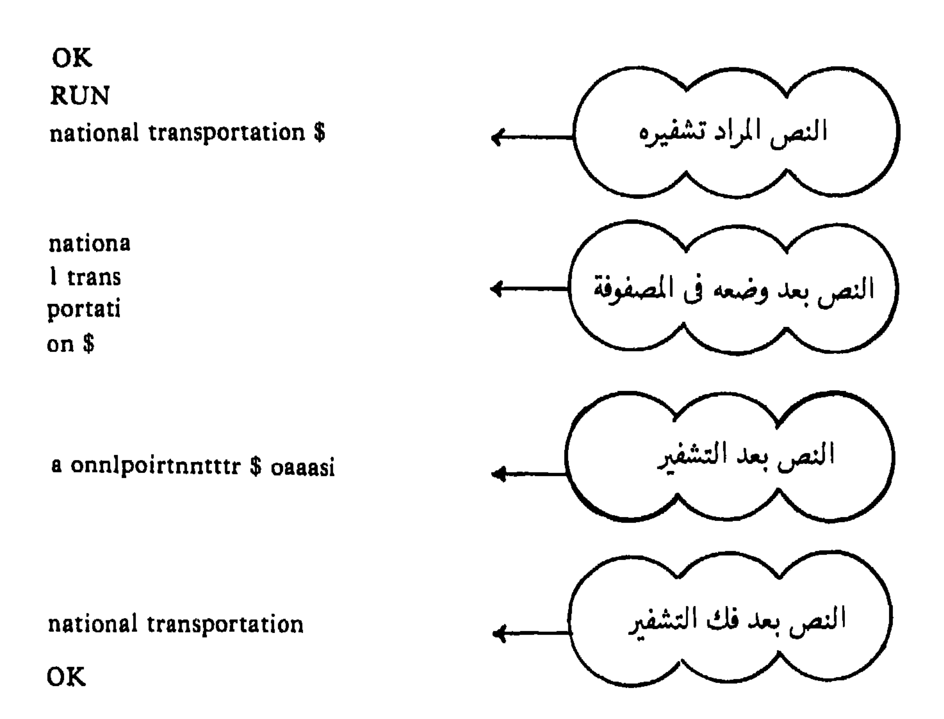
شكل (١٥ ــ ٣)

برنامج لتحويل نص معين إلى نص الشفرة باستخدام طريقة الأعمدة

```
10 DIM A$(20,20)
20 REM  استدعاء برنامج فرعي
30 GOSUB 70
40 REM  استدعاء برنامج فرعي
50 GOSUB 310
60 END
70 REM برنامج فرعي لقراءة ابعاد المصفوفه و النص
80 READ R,C
90 READ R$
100 PRINT R$ :REM  طباعة النص المدخل
110 PRINT
120 LET S = 0
130 FOR I = 1 TO R
140 FOR J = 1 TO C
150 LET S = S + 1
160 LET B$ = MID$(R$,S,1)
170 IF B$ = "$" THEN A$(I,J) = B$
180 IF B$ = "$" THEN 220
190 LET A$(I,J) = MID$(R$,S,1)
220 NEXT J
230 NEXT I
240 FOR I = 1 TO R
```

```
250 FOR J = 1 TO C
255 REM   طباعة النص من المصفوفه
260 PRINT A$(I,J)
270 NEXT J
280 PRINT
290 NEXT I
300 RETURN
310 REM
320 FOR I = 1 TO C
330 READ P(I):NEXT I
340 FOR I = 1 TO C
350 LET J = P(I)
360 FOR K = 1 TO R
365 REM   طباعة النص المشفر
370 PRINT A$(K,J);
380 NEXT K
390 NEXT I
400 PRINT : PRINT : PRINT
410 FOR I = 1 TO R
420 FOR J = 1 TO C
430 IF A$(I,J) = "$" THEN 460
435 REM   طباعة النص بعد اعادة تشفيره
440 PRINT A$(I,J);
450 NEXT J: NEXT I
460 REM
470 RETURN
480 DATA 5,7
490 DATA "national transportation$"
500 DATA 2,1,4,6,3,5,7
510 REM
```

وفى حالة تنفيذ البرنامج فى الشكل السابق سنحصل على النتائج التالية :



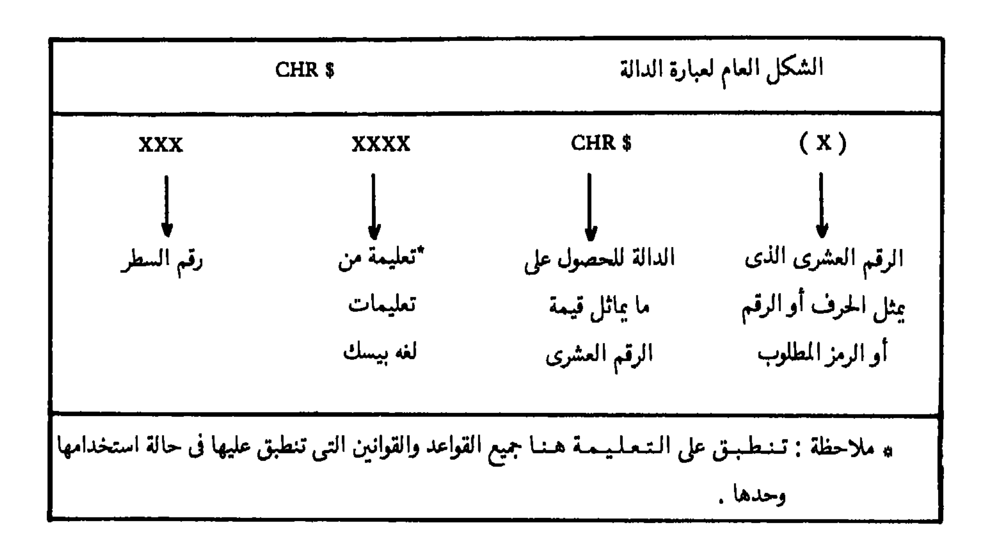
تعليمات أخرى للتعامل مع السلسلات

التـعـامل مع الأ رقام الممثلة للحروف والأ رقام والرموز داخل الحاسب : يتم تمثيل البـيانات داخل الحاسب الآلى باستخدام النظام الثنائى (SYSTEM BINARY) وفق أنـظـمـة مختـلـفـة ، ومـن أشهـر هذه الأنظمة والتى تستخدم فى معظم الحاسبات الآلية الشخصية هو النظام الأمريكى المعيارى لتبادل المعلومات :

(AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE)

واختصاره (آسكى ASCII)

حيـث يـتـم تمـثيل كل من الحروف والرموز الخاصة والأ رقام العشرية بعدد عشرى يسـهـل عـلـيـنـا قراءته . ويمكننا الحصول على الحرف أو الرقم أو الرمز باستخدام الدالة CHR$ كما هو مبين بالشكل التالى :[1]



(١) الجـدول الـذى يـبين الحـروف والأ رقـام والـرموز التى يمكن الحصول عليها وما يماثلها فى كل من النظامين (العشرى Decimal) و (السداسى عشر Hexadecimal) يمكن إيجاده فى دليل اللغة الخاص بالحاسب المستخدم .

فمثلاً لو أردنا طباعة الحروف (الكبيرة Capital) الأ بجدية اللا تينية نستخدم العبارات التالية :

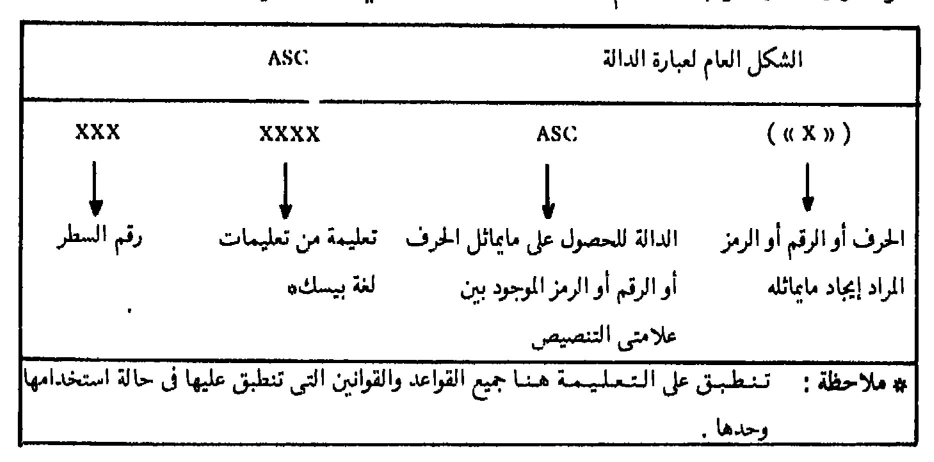
```
10  FOR I = 65 TO 90
20  PRINT CHR $ (I);
30  NEXT I
40  END
```

ملاحظة : تم استخدام هذه العبارات مع نظام التشغيل (DOS 3.2) .

وفى حالة تنفيذ هذه العبارات سنحصل على النتائج التالية :

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

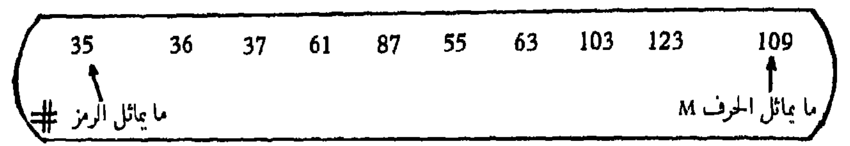
كما أمكننا الحصول على الحروف والأ رقام والرموز باستخدام الدالة CHR $ وما يماثل الحرف المراد بالنظام العشرى ، يمكننا الحصول على القيمة العشرية التى تماثل حرفاً أو رقماً أو رمزاً باستخدام الدالة ASC كما هو مبين بالشكل التالى :



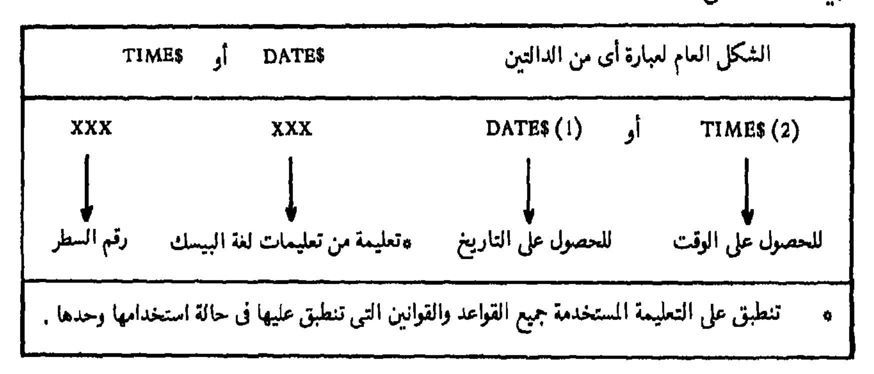
فمثلا لو أردنا إيجاد مايماثل كلاً من الحروف والأرقام والرموز التالية نستخدم العبارات التالية :

```
10   FOR I = 1 TO 10
20   READ A$
30   PRINT ASC («A$»); «       » ;
40   NEXT I
50   DATA «#», «$», «%», «=», «W», «7», «?», «g», «(», «m»
60   END
```

وفى حالة تنفيذ هذه العبارات سنحصل على النتائج التالية :



الحصول على التاريخ والوقت من الحاسب : يمكننا الحصول على التاريخ والوقت من ذاكرة الحاسب الرئيسية باستخدام الدالتين DATE$ و TIME$ كما هو مبين فى الشكل التالى :



١ ـــ ستكون الصيغة للتاريخ :   السنة   /   اليوم   /   الشهر

مكون من : أربع خانات   خانتين   خانتين = ١٠ خانات

٢ ـــ ستكون الصيغة للوقت :   الثانية   /   الدقيقة   /   الساعة

مكون من : خانتين   خانتين   خانتين = ٨ خانات

فمثلاً لو أردنا الحصول على السنة والدقيقة نستخدم العبارات

```
10    LET Y$ = RIGHT$ (DATE$, 4)
20    LET M$ = MID$ (TIME$, 4, 2)
30    PRINT M$,«ودقيقة»; Y$; «سنة»
40    END
```

وفى حالة تنفيذ هذه العبارات سنحصل على النتائج التالية :

سنة 1987 ودقيقة 15

كما يمكننا تغيير الوقت والتاريخ متى نشاء، وذلك بإسناد القيم المرادة لكل من الدالتين مع مراعاة الصيغة العامة لكل منهما كما يلى :

```
10    LET DATE$  =« 10/12/1986»
20    LET TIME$  =« 08:30:10 »
```

ففى العبارة رقم 10 أصبح التاريخ فى الذاكرة الرئيسية للحاسب يساوى الثانى عشر من شهر أكتوبر لسنة ست وثمانين وتسعمائة و ألف .

وفى العبارة رقم 20 أصبح الوقت فى الذاكرة الرئيسية للحاسب يساوى الساعة الثامنة والنصف وعشر ثوان .

الحصول على البيانات المرحلة من لوحة الأزرار دون تحديد اسم متغيرها :

تعرضنا فى السابق إلى كيفية الحصول على البيانات بطرق مختلفة سواء بواسطة تعليمة أدخل INPUT أو اقرأ READ أو الإسناد LET وعلى ضوء نوعية هذه البيانات كان يتم تحديد اسم المتغير الذى ستسند إليه (متغير عددى للبيانات العددية أو متغير غير عددى للبيانات غير العددية) .

وسنتعرض الآن للدالة INKEY$ حيث تمكننا فى الحصول على البيانات بواسطة لوحة المفاتيح حرفاً أو رقماً أو رمزاً فى كل مرة نستخدمها دون إظهار هذه القيمة على شاشة الجهاز كما هو مبين فى الشكل التالى :

الشكل العام لعبارة الدالة		INKEY$	
= INKEY$	X$	LET	XXX
↓	↓	↓	↓
الدالة لإشعار الحاسب باستقبال أى مفتاح من لوحة المفاتيح دون إظهار القيمة على الشاشة	اسم المتغير الذى ستسند له قيمة المفتاح الذى سيضغط	تعليمة أسند لإسناد قيمة المفتاح للمتغير X$	رقم السطر

كيفية عمل الدالة INKEY$ متشابهة إلى حد كبير مع كيفية عمل التعليمة INPUT إلا أنهما تختلفان فى :

١) تحديد اسم المتغير ، ففى الدالة INKEY$ تسند القيمة إلى متغير غير عددى دائماً ،

أما فى تعليمة أدخل INPUT فيتم استخدام اسم متغير على ضوء نوعية البيانات المراد إدخالها .

٢) باستخدام تعليمة أدخل INPUT يبقى الجهاز منتظراً إلى أن يتم إدخال القيمة ، أما فى دالة INKEY$ فلن ينتظر الجهاز إدخال القيمة المرادة .

وللحيلولة دون ذلك يمكننا استخدام العبارات التالية :

```
10    LET A$ = INKEY$
20    IF A$  = «  » ,THEN 10
```

هنا تم وضع الجهاز فى حالة دوران LOOP طالما أنه لم يتم إدخال قيمة (الضغط على أى من المفاتيح) .

فمثلا لو أردنا الاستفسار عن كلمة السر للتعامل مع برنامج معين للتأكد من السماح للمستخدم بالتعامل مع هذا البرنامج ، نعمل على إسناد كلمة السر لمتغير ومن ثم نستخدم الدالة INKEY$ للحصول على كلمة السر ، حيث سيتم إدخالها دون إظهارها على الشاشة وهذا يقلل/يمنع من تسربها ، و يتم ذلك باستخدام العبارات التالية :

```
10  P$ = «P302» : REM P$                إسناد كلمة السر للمتغير
20  LET L = LEN (P$) : REM              للحصول على طول كلمة السر
30  FOR I = 1 TO L
                                        للحصول على كلمة السر حرفاً حرفاً
40  LET MID$ (E$,I,1) = INKEY$ : REM
                                        إذا لم يتم الضغط على مفتاح ارجع إلى تعليمة
50  IF MID$ (E$,I,1) = «  » THEN 40 : REM 40
60  NEXT I
```

```
70 IF E$ = P$ THEN 100 : REM للاستفسار عن صحة كلمة السر
80 PRINT «XXXX خطأ فى كلمة السر لايسمح لك بالتعامل مع البرنامج XXX»
90 END
100 تكملة البرنامج
```

تمارين

١ ـــ اعمـل على طباعة وتنفيذ البرامج الموجودة فى أشكال الفصل ، وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعديلات .

٢ ـــ بين الأخطاء إن وجدت فى كل من العبارات التالية :

أ `10 LET A$ = B$ + N + N$`

ب `10 LET A$ = LEFT («ALI» ,1)`

جـ `10 LET A$ = RIGHT («ALI» ,4)`

د `10 LET A$ = (B$)`

هـ `10 LET A$ = VAL (LEFT(«ALI» ,3))`

و `10 PRINT CHAR$ (20)`

ز `10 LET A$ = MID («ALI» ,2,2)`

٣ ـــ اكتب برنامجاً يعمل على قراءة النص التالى :

A COMPUTER IS A DATA PROCESSING MACHINE.

ومن ثم إيجاد التالى :

أ) عدد المرات التى يتكرر بها كل حرف من الحروف الأ بجدية اللا تينية .

ب) عدد الكلمات التى يتكون منها النص .

جـ) عدد الجمل التى يتكون منها النص .

د) طول النص ، عدد الحروف (CHARACTERS) .

٤ ــ يتكون رمز الشقة فى مجمع كبير من الحقول التالية :

١ ــ مستوى الشقة وعدد غرف النوم لكل نوع (خانتان) .

النوع	عدد غرف النوم
ممتاز ــ D	٤ ــ ٦
متوسط ــ S	٢ ــ ٤
عادى ــ N	١ أو ٢

٢ ــ عدد الأ بواب (خانتان) .

٣ ــ عدد الشبابيك (خانتان) .

٤ ــ الموقع على بركة ماء صناعية (خانة واحدة) .

صفر ــ لا ١ ــ نعم .

٥ ــ نوع المستأجر (خانة واحدة) . غير مؤجر صفر

شخصى لفرد ١

شخصى لعائلة ٢

شركة ٣

٦ ــ رقم الشقة (ثلاث خانات)

طور برنامجاً لحساب التالى :

١ ــ عدد الشقق غير المؤجرة لكل نوع .

٢ ــ عدد الشقق المؤجرة حسب نوع المستأجر .

٣ ــ عدد الأ بواب والشبابيك فى المجمع لأغراض الصيانة .

٤ ــ عدد الشقق غير المؤجرة والموجودة على بركة الماء .

من الأمثلة على البيانات المدخلة :

D54600112

S33512433

N13401315

٥ ــ يتكون رمز الحجرة فى المستشفى المركزى من الحقول المرمزة التالية :

١) نوع الحجرة (خانة واحدة) P خاص

S شبه خاص G عام

٢) إذا كان غير خاص ، فعدد الأسرة فى الحجرة (خانة واحدة) . شبه الخاص يكون العدد ٢ أو ٤ و العام ٤ ــ ٧ أسرة .

٣) هل الحجرة فيها هاتف أم لا (خانة واحدة) T مع هاتف N دون هاتف .

٤) هل الحجرة فيها تلفاز (خانة واحدة) V مع تلفاز N دون تلفاز .

٥) هل الحجرة فيها تكييف خاص أو تابعة C مركزى للتكييف المركزى D محلى

٦) رقم الطابق (خانة واحدة) هناك ثلاثة طوابق فى المستشفى .

٧) رقم الحجرة (خانتان) .

٨) مؤشر للدلالة عن مدى شغور الأسرة فى الحجرة 0 دون شواغر 7 - 1 عدد الشواغر .

المطلوب :

١ ــ أوجد عدد الأسرة الموجودة فى الحجرات الخاصة وشبه الخاصة والعامة .

٢ ــ أوجد عدد الأسرة الشاغرة فى كل نوع من أنواع الحجر .

٣ ــ أوجد عدد الأسرة فى الحجرات (شبه الخاصة أو العامة) وفى كل منها هاتف .

٤ ــ أوجد عدد الأسرة فى الحجرات التى فى الطابق الأول والتى فى كل منها هاتف، وخاصة.

٥ ــ أوجد عدد الأسرة فى الحجرات التى فى كل منها هاتف وتلفاز.

٦ ــ أوجد عدد الأسرة فى الحجرات الخاصة والتابعة للتكييف المركزى .

٧ ــ أوجد عدد الأسرة فى الحجرات التى فى كل منها (تلفاز أو هاتف) و (تقع فى الطابق الأول أو الثانى) .

من الأمثلة على المدخلات :

P1TND2140

C4NNC3153

S2TVC2162

الفصل السادس عشر

الرسومات البيانية

مقدمة عن الرسومات البيانية

فى الفصول السابقة ، تم استخدام الأرقام والجداول لعرض المعلومات إما على الشاشة أو على الورق ، ومع أن استخدام الرسومات البيانية لعرض المعلومات قديم عهد فى الإحصاء ، إلا أنه حديث عهد فى الحاسب .

وتمتاز لغة البيسك عن غيرها من اللغات بتوفر تعليمات برمجة تساعد فى عمل رسومات على الشاشة . ومن أنواع هذه الرسومات التى يمكن رسمها فى لغة البيسك الدوائر المقسمة والخطوط والبيانات والأعمدة .

فوائد استخدام الرسومات البيانية

ومن فوائد استخدام الرسومات لإظهار نتائج المعالجات بدلاً من القيم والجداول :

١ ــ اختصار الوقت فى الاطلاع على نتائج التقارير وخاصة للمديرين الذين ليس لديهم الوقت الكافى لفعل ذلك ، فالرسومات البيانية يسهل الاطلاع عليها بطريقة أسرع وأجدى .

٢ ــ سهولة استنتاج الحالات الغريبة والخارجة عن النمط العام للبيانات ، كزيادة مبيعات شهر رجب عن غيره من الشهور.

٣ ــ سهولة المقارنة بين البيانات التابعة لظواهر مختلفة ، والتى تحتوى فى معظم الحالات على بيانات كثيرة ، كالمقارنة بين مبيعات مناطق مختلفة تابعة لشركات متعددة .

ــ سهولة استنباط نمط عام (اتجاه) للبيانات ، مثل ملاحظة الازدياد المطرد للمبيعات خلال السنوات الخمس الماضية.

٥ ـ إمكانية استخدام الألوان لتسهيل التوصل إلى النتائج السابقة .

وقبل تطوير برامج للرسومات البيانية ، سنتعرض لأوضاع شاشة العرض وكيفية التحكم فيها .

أوضاع الشاشة

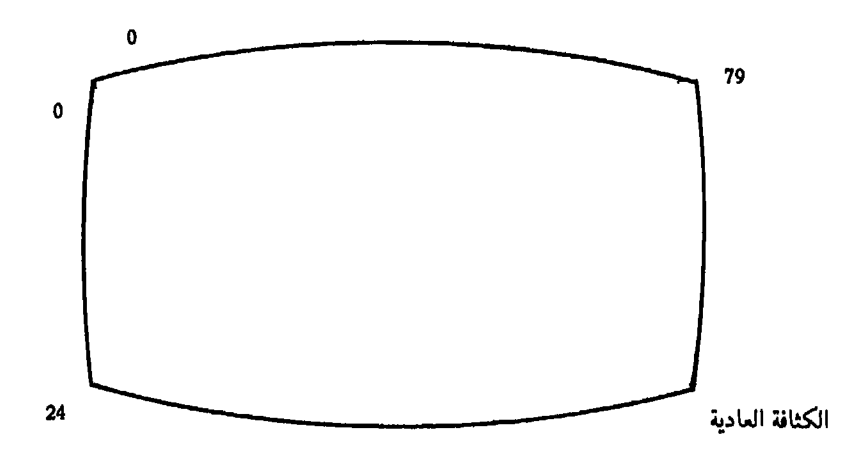
للشاشة ثلاثة أوضاع تحدد عدد الأسطر والأعمدة فيها : كلما زاد عدد الأسطر والأعمدة ارتفعت كثافة الشاشة وقلت إمكانية استخدام الألوان . ولتحقيق الكثافة فوق العادية على الحاسبات الشخصية ، لابد من وجود شاشة ملونة خاصة بالرسومات ، أما الشاشات العادية ، ففيها الكثافة العادية فقط . و يوضح الجدول التالى أوضاع الشاشة :

نوع الشاشة	كثافة الشاشة عادية TEXT	SCREEN RESOLUTION متوسطة MEDIUM	عالية HICH
	٨٠ عموداً ٢٥ سطراً	٣٢٠ عموداً ٢٠٠ سطر	٦٤٠ عموداً ٢٠٠ سطر
عادية	١٦ لوناً	غير متاح	غير متاح
ملونة خاصة بالرسومات	١٦ لوناً	٤ ألوان	لونان أبيض وأسود

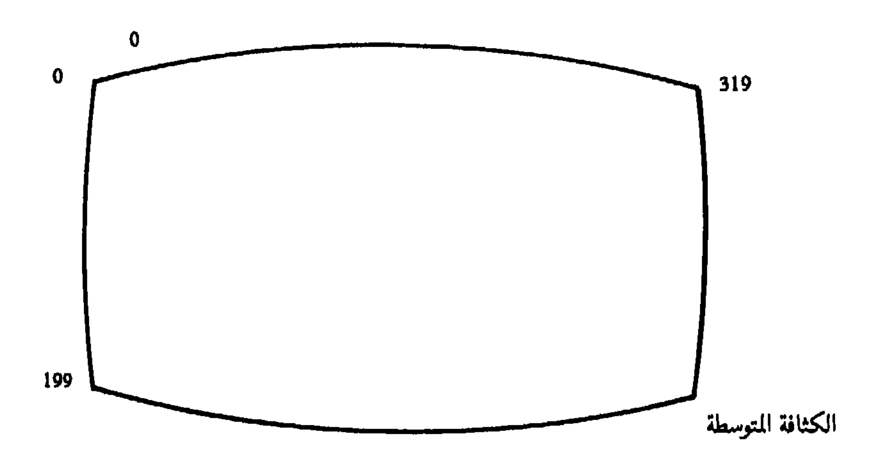
وتتحدد الكثافة بمقدار حجم الخلية المستخدمة للرسم البيانى والناتجة عن تقاطع السطر والعمود و يوضح ذلك الشكل التالى :

	الكثافة العادية	الكثافة المتوسطة	الكثافة العالية
	٨٠ × ٢٥	٣٢٠ × ٢٠٠	٦٤٠ × ٢٠٠
مجموع الخلية الواحدة النسبى	☐	☐	▭

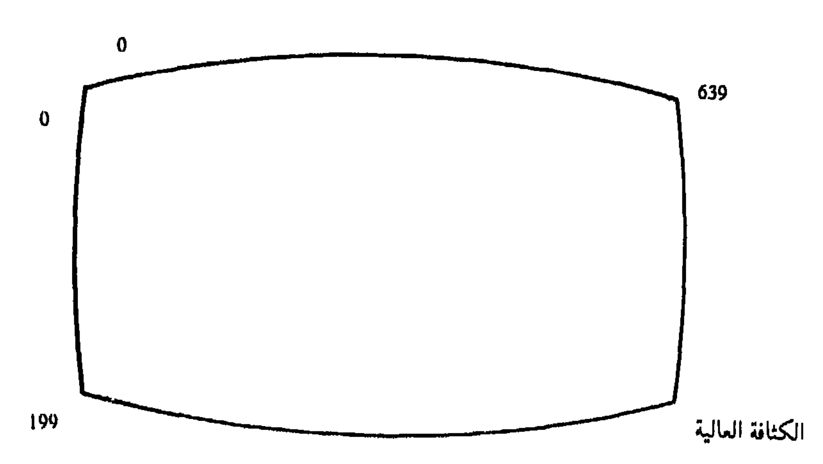
ويمكن تصوير حجم هذه الخلايا على الشاشة ذات البعدين الثابتين كما هوموضح في الأشكال التالية :



الكثافة العادية



الكثافة المتوسطة



لاحظ أن الكثافة العادية تستخدم بشكل رئيسى فى حالة كتابة النصوص والبرامج ، وهناك الكثافة تحت العادية (٤٠ عموداً و ٢٥ سطراً) ، والتى تستخدم أحياناً لطباعة الحروف والكلمات بالحجم الكبير ، وذلك لتسهيل الرؤية والقراءة والمتابعة للعمليات المصاحبة لتنفيذ البرنامج .

التحكم فى أوضاع الشاشة

ولوضع الشاشة فى أى من الحالات الثلاث ، واعتماداً على توفر الشاشة الملونة من عدمه تستخدم تعليمة «الشاشة SCREEN» ، كما هو مبين فى الشكل التالى :

الشكل العام لتعليمة		SCREEN
N	SCREEN	XXX
↓	↓	↓
* متغير	التعليمة	رقم السطر

* ملاحظة : يأخذ المتغير N أحد الأرقام التالية :

0 : الكثافة العادية ٨٠ عموداً و ٢٥ سطراً

1 : الكثافة المتوسطة ٣٢٠ عموداً و ٢٠ سطراً

2 : الكثافة العالية ٦٤٠ عموداً و ٢٠٠ سطر

وهنـاك تـوصيـفـات إضـافية للشاشة يمكن الحصول عليها من دليل اللغة الخاص بكل جهاز ، وأوردنا هنا مايهمنا فقط فى هذا الفصل .

تطوير برامج على الرسومات البيانية

سنقـدم فى هذا القسم عدة أمثلة عن كيفية استخدام تعليمات الرسومات البرمجية لعمل رسومات تمثل نتائج المعالجات فى البرنامج .

مثال : (١٦ ـــ ١) : (الأعمدة) .

الهدف : إظهار مبيعات شركة معينة خلال الا ثنى عشر شهراً للسنة الماضية .

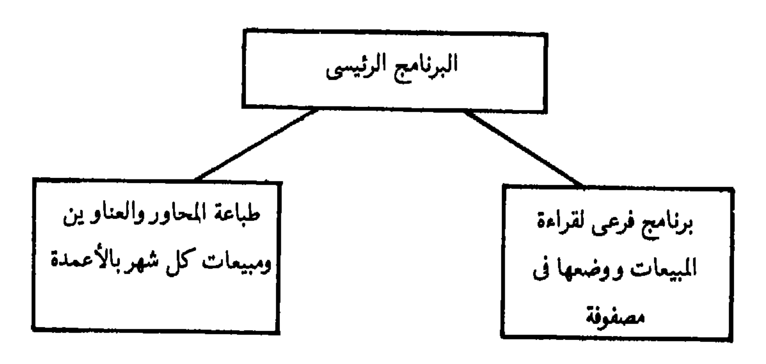
أولاً ـــ خطوات الحل :

١ ـــ تحديد محاور الشهور والمبيعات وطباعتها .

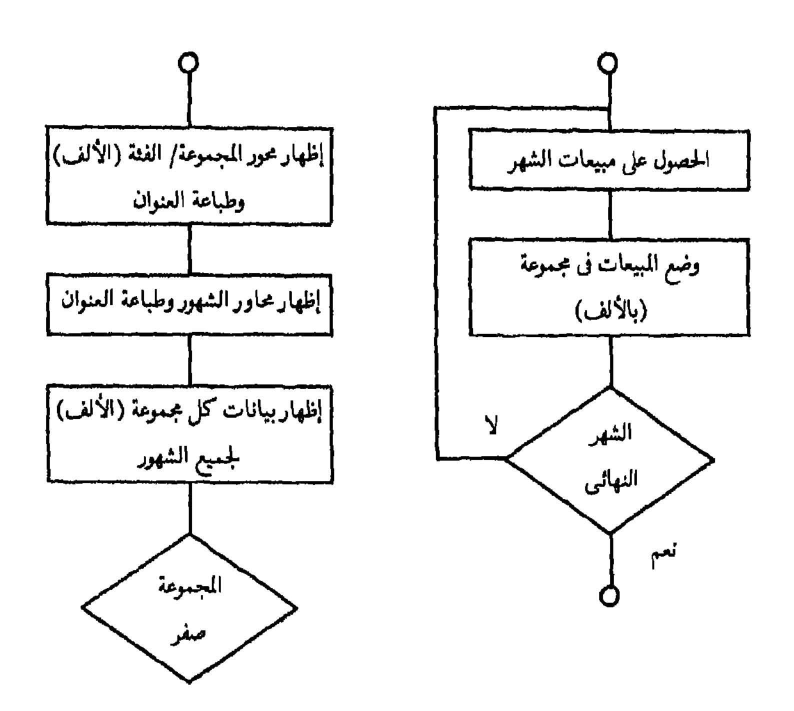
٢ ـــ الحصول على مبيعات كل شهر ووضعها فى مصفوفة وفئات (بالألف ريال) .

٣ ـــ تمثيل المبيعات فى أعمدة وطباعتها .

ثانياً ـــ الهيكل الهرمى :



ثالثاً ــ نمط البرمجة التركيبية :



رابعاً ــ البرنامج فى الشكل التالى :

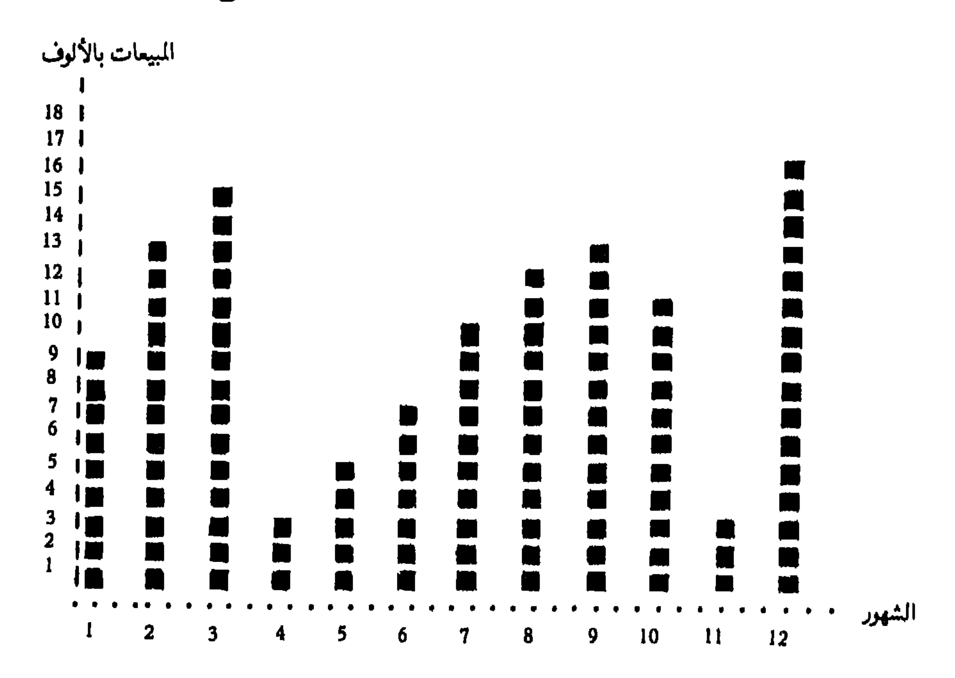
شكل (١٦ ــ ١)

برنامج لإظهار المبيعات الشهرية على شكل أعمدة

```
10 DIM A(20),B(20): KEY OFF: CLS
20 GOSUB 50: REM   استدعاء برنامج فرعي لقراءة المبيعات لكل شهر
30 GOSUB 120:REM   استدعاء برنامج فرعي للا ظهار المبيعات لكل شهر
40 END
50 REM      برنامج فرعي لقراءة مبيعات كل شهر و وضعها في مصفوفه
60 FOR I=1 TO 12
70 READ A(I)
80 REM  لوضع مبيعات كل شهر في مجموعات بالا لف
90 LET B(I) = A(I) / 1000
100 NEXT I
110 RETURN
120 REM  برنامج فرعي لا ظهار مبيعات كل شهر بالرسومات ( بالا عمده)
130 REM  لطباعة العناوين و محور المبيعات
140 LOCATE 1,1: PRINT "المبيعات بالا لوف"
150 FOR I = 21 TO 2 STEP -1
160 LOCATE I,10: PRINT CHR$(219)
170 NEXT.I
180 REM      لطباعة العناوين و محور الا شهر
190 FOR I = 10 TO 60
200 LOCATE 21,I: PRINT CHR$(176);
```

```
210 NEXT I
220 FOR I = 1 TO 12
230 LOCATE 22,10 + (I - 1) * 4: PRINT USING "##";I;
240 NEXT I
250 LOCATE 22,10 + (I - 1) * 4: PRINT "شهر ٪!"
260 REM  لا ظهار مبيعات كل شهر
270 FOR I = 1 TO 12
280 FOR J = 20 TO.30 - 2 * B(I) STEP -1
285 LET Y = 11 + ( I - 1 ) * 4
290 LOCATE J,Y
300 PRINT CHR$(132);
310 NEXT J
320 NEXT I
330 RETURN
340 DATA 09000,11000,12000,6000,07000,8300
350 DATA 9500,10600,11300,10000,6400,12700
```

وفى حالة تنفيذ البرنامج فى الشكل السابق، سنحصل على النتائج التالية :



كما نلاحظ من نتائج المعالجة (فى كلا المثالين ١٠ ــ ١، ١٠ ــ ٢) أنه من السهل ملاحظة وإيجاد الشهر الذى تمت فيه أعلى قيمة مبيعات، سواء كان ذلك للشركة أو لأى من المندوبين الثلاثة، وأن الوقت المستغرق لإيجاد أفضل شهر أو أسوأ شهر أقل بكثير من الوقت الذى سيستغرق فى حالة كون نتائج المعالجة (المخرجات) رقمية .

مثال (١٦ ــ ٢) : (النقاط المبعثرة Scatterdiagram)

الهدف : إظهار طول المسافة التى يحتاج إليها السائق لإيقاف سيارته على حسب السرعة التى يسير بها والتى تتراوح ما بين ٢٠ إلى ٨٠ كيلومتراً فى الساعة .

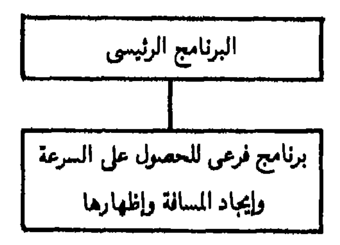
أولاً ـ خطوات الحل :

١ ـ الحصول على السرعة .

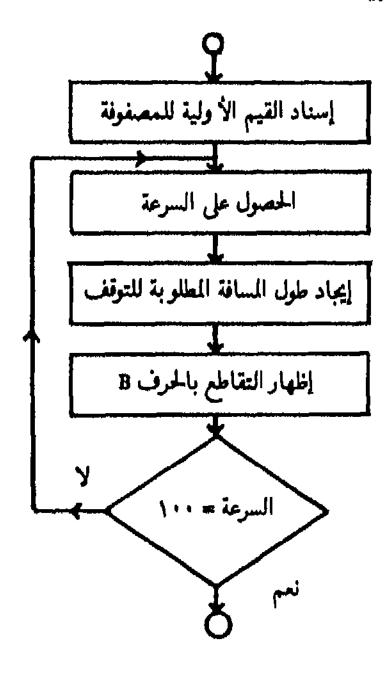
٢ ـ إيجاد طول المسافة .

٣ ـ إظهار تقاطع السرعة والمسافة بالـ (*) .

ثانياً ـ الهيكل الهرمى :



ثالثاً ـ نمط البرمجة التركيبية :



رابعاً ــ البرنامج فى الشكل التالى :

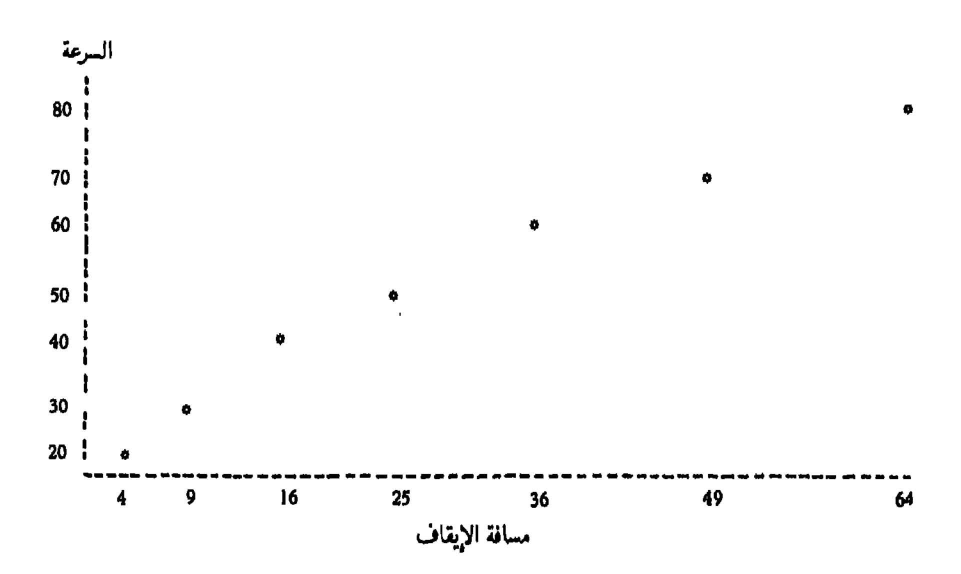
شكل (١٦ ــ ٢)

برنامج لإظهار المسافة اللازمة لإيقاف السيارة حسب سرعتها

```
10 CLS
20 SCREEN 2,1,0
30 DIM T(20),S(12),P(12),A(12)
40 GOSUB 70  :REM  أستدعاء برنامج فرعي لقراءة البيانات
50 GOSUB 210 :REM  أستدعاء برنامج فرعي لطباعة المخرجات
60 END
70 REM  لقراءة البيانات ووضعها في مصفوفه وايجاد النسبه المئويه لمبيعات كل الشهر
برنامج فرعي
80 LET T=0
90 FOR I =1 TO 12
100 READ S(I)
110 LET T = T + S(I) :REM  أضافة مبيعات الشهر لمجموع المبيعات
120 NEXT I
130 REM  للا يجاد النسبه المؤيه لمبيعات كل شهر
140 FOR I = 1 TO 12
150 LET P(I) = (S(I) / T)
160 NEXT I
170 REM  البيانات
180 DATA 6000,10000,13000,7500,10500,14000
190 DATA 9300,3500,10000,5000,11000,14000
```

```
200 RETURN
210 REM  برنامج فرعي للاظهار مبيعات كل شهر بالرسم الدائري
220 FOR I=0 TO 12 : REM  للاسناد القيمه الاوليه لزاوية كل شهر
230 LET A(I) = 0
240 NEXT I
250 REM  للايجاد الزاويه لمبيعات كل شهر
260 FOR I = 1 TO 12
270 LET D = P(I) * (2 * 3.141)
280 LET A(I) = A(I - 1 ) + D
290 CIRCLE (400,100),80,1,-A(I),-A(I-1)
300 NEXT I
310 REM  لطباعة النسبه المؤيه لمبيعات كل شهر
320 LOCATE 1,1: PRINT "الشهر     النسبه المؤية لمبيعاته "
330 LOCATE 2,1: PRINT "======================    ====="
340 FOR I=1 TO 12
350 PRINT USING      "        #.###                ## ";P(I),I
360 NEXT I
370 RETURN
```

وفى حالة تنفيذ البرنامج، ستظهر النتائج التالية :



مقدمة عن استخدام الألوان فى الرسومات

للحصول على نتائج المعالجة بالألوان يشترط أن يكون للجهاز إمكانية عمل ذلك بوجود شاشة ملونة .

ففى جهاز IBM PC/XT يمكننا الحصول على الألوان التالية فى حالة الكثافة العادية :[1]

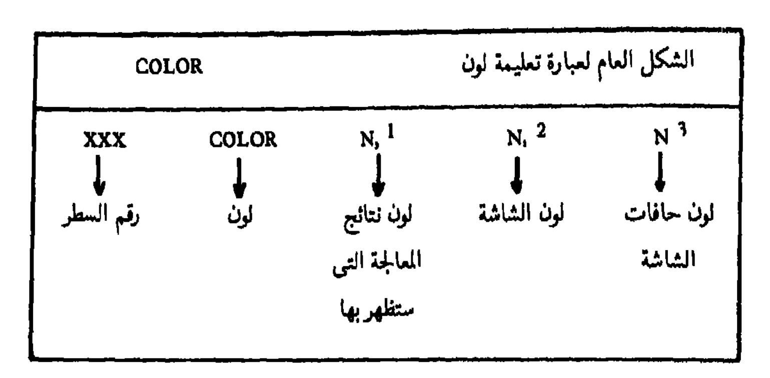
رمز اللون	نوعه	رمز اللون	نوعه
0	BLACK أسود	8	GRAY رمادى
1	BLUE أزرق	9	LIGHT BLUE أزرق فاتح
2	GREEN أخضر	10	LEGHT GREEN أخضر فاتح
3	CYAN أزرق داكن	11	LIGHT CYAN أزرق فاتح
4	RED أحمر	12	LIGHT RED أحمر فاتح
5	MAGNETA بنفسجى	13	LIGHT MAGNETA بنفسجى فاتح
6	BROWN بنى	14	YELLOW أصفر
7	WHITE أبيض	15	HIGH - INTENSITY WHITE أبيض ناصع

وأما فى حالة اختيار الكثافة المتوسطة فيكون اختيار الألوان كالتالى :

(0 ــ أسود، 1 ــ أزرق، 2 ــ بنفسجى، 3 ــ أبيض) أما الكثافة العالية فيكون اختيار اللونين (0 ــ أسود و 1 ــ أبيض) .

ولاختيار اللون نستخدم تعليمة لون COLOR كما هو مبين بالشكل التالى :

(1) IBM Personal Computer Hardware Reference Library BASI C, p. 39



١ ـــ و يطلق عليه FOREGROUND ويجب أن يكون الرقم ما بين (صفر ـــ ١٥) .

٢ ـــ و يطلق عليه BACKGROUND ويجب أن يكون الرقم ما بين (صفر ـــ ٧) .

٣ ـــ و يطلق عليه BORDER ويجب أن يكون الرقم ما بين (صفر ـــ ١٥) .

تطوير برنامج باستخدام تعليمة لون COLOR

يمكن تعديل البرنامج السابق وعرض الرسومات بالألوان كالتالى :

إذا أردنا عرض البيانات بالألوان، استخدمنا العبارة التالية فى شكل (١٦ ـــ ٢) :

```
25  COLOR  0, 4, 2
```

فنجد أن المخرجات ستظهر باللون الأسود (٠) ولون الشاشة سيكون أحمر (٤) ولون حافات الشاشة سيكون أخضر (٢) .

وهنا سيستمر عرض جميع المخرجات بهذه الألوان إلى أن يتم استخدام تعليمة لون COLOR مرة أخرى فى البرنامج .

فلعرض المخرجات بالألوان للمندوبين الثلاثة (العملية رمز ٥) فى شكل (١٦ ـــ ٢) نستخدم العبارة التالية :

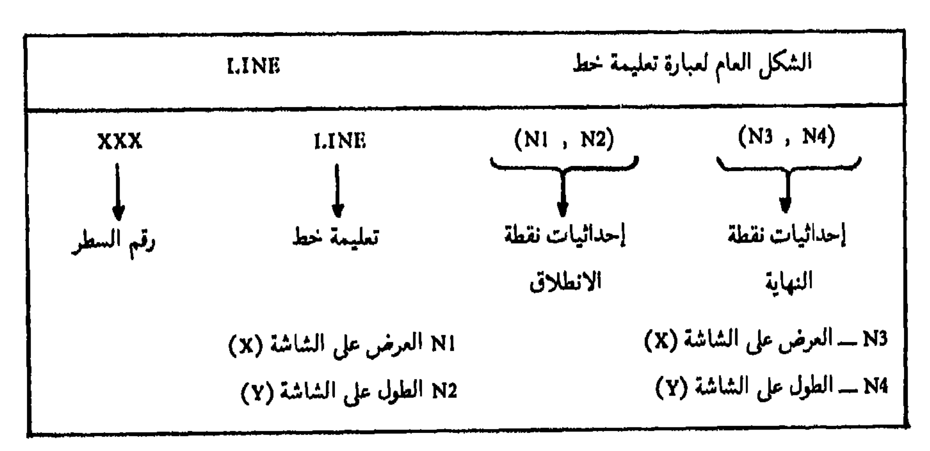
```
1215  COLOR J- 1, 14, 12
```

وفى حالة تنفيذ البرنامج ستظهر مبيعات المندوب الأول باللون الأسود، ومبيعات المندوب الثانى باللون الأزرق، ومبيعات المندوب الثالث باللون الأخضر، وسيكون لون الشاشة فى جميع الحالات أصفر (١٤)، ولون الحافات أحمر فاتحاً (١٢)* .

تعليمات رسم الأشكال الهندسية مسبقة التحديد

للحصول على الأشكال الهندسية المختلفة كالدوائر والمربعات والمثلثات والخطوط نستخدم التعليمات التالية فى حالتى الكثافة المتوسطة والعالية :

الخط المستقيم : إذا أردنا رسم خط نستخدم تعليمة خط LINE كما هو مبين بالشكل التالى :



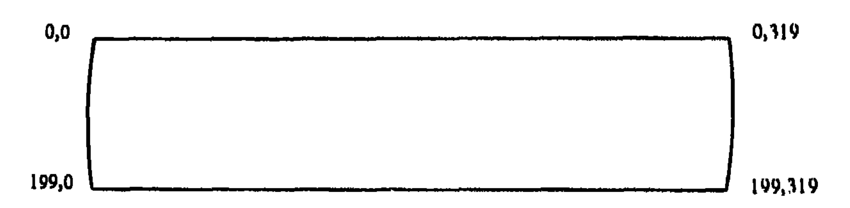
إن أى نقطة على الشاشة تمثل بقيمتين و يطلق عليهما «إحداثيات»، الأولى تحدد العرض (X) أى العمود Column، أما الثانية فتحدد الطول (Y) أى السطر ROW وهذه الإحداثيات تمثل ما يطلق عليه اسم خلية PIXEL .

* ملاحظة : للحصول على مزيد من توصيفات تعليمة لون COLOR يمكن العودة إلى دليل لغة بيسك الخاص بالجهاز المستخدم .

فبالعبارتين التاليتين :

```
10  SCREEN 1
20  LINE (12,100) - (100,150)
```

نعمل على رسم خط مستقيم يبدأ من النقطة التى إحداثياتها (12,100) و ينتهى فى النقطة التى إحداثياتها (100, 150) ، أى أن المؤشر انطلق من مركز الإحداثيات الأولى إلى اليمين بثمان وثمانين PIXEL وإلى الأسفل بخمسين PIXEL كما هو مبين فيما يلى :



ويمكننا رسم خط آخر مرتبط بالخط الذى انتهينا من رسمه باستخدام تعليمة خطوة STEP كما هو مبين بالشكل التالى :

الشكل العام لعبارة تعليمة خطوة		STEP	
(N1, N2)	STEP	LINE -	XXX
إحداثيات النقطة المراد تكملة رسم الخط لها	تعليمة خطوة	تعليمة خط	رقم السطر

فإذا أردنا تكملة رسم الخط الذى انتهت إحداثياته بالقيمتين (100,150) بخمسين وحدة إلى اليمين وعشرين وحدة إلى أعلى ، نستخدم العبارة التالية :

```
30 LINE - STEP (50,20)
```

فإذا أردنا أن يكون لون الخط الجديد أزرق فستصبح العبارة كالتالى :

```
30 LINE - STEP (50, - 20), 1
```

وعليه سيصبح الخط الجديد كالتالى :

الخط الجديد باللون الأ زرق

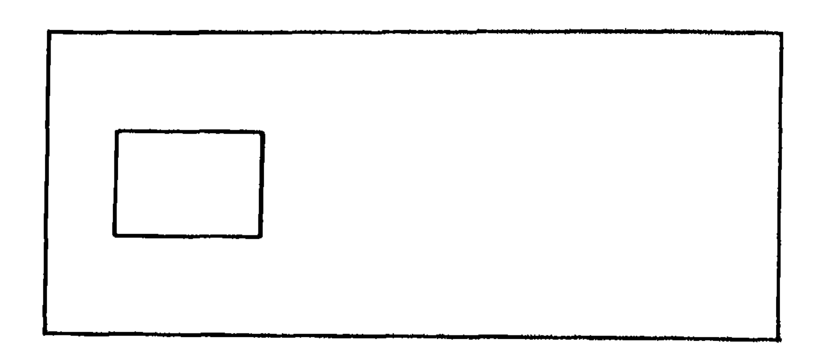
وباستخدام تعليمة خط LINE يمكننا رسم أشكال رباعية ، حيث إن كل ما هو مطلوب لرسم الأشكال الرباعية تحديد إحداثيات زاويتين (نقطتين) متعاكستين ، واستخدام حرف B اختصاراً لكلمة صندوق BOX مقروناً بنهاية تعليمة خط ، كما هو مبين بالشكل التالى :

الشكل العام لعبارة تعليمة خط LINE لرسم أشكال رباعية						
XXX	LINE	(N1, N2)	–	(N3, N4)	N,	B
رقم السطر	تعليمة خط	إحداثيات الزاوية الأولى		إحداثيات الزاوية المعاكسة	رمز لون خطوط الشكل	لتحديد أن الشكل رباعى

فبالعبارة التالية

```
10 LINE (10,30) - (70,100), 1, B
```

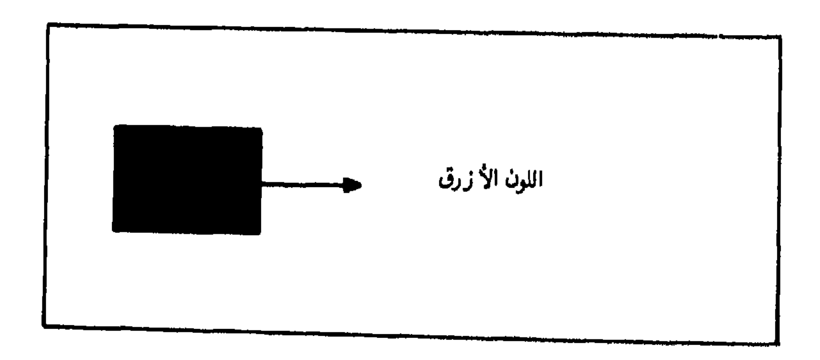
نعمل على رسم شكل رباعى وإحداثيات زاويته الأولى (10, 30)، والزاوية المعاكسة (70, 100)، ولون الخطوط سيكون أزرق كما هو مبين فيما يلى :



كما يمكننا تلوين هذا الشكل الرباعى بكامله من الداخل باستخدام حرف F اختصاراً لكلمة FILLED ملاصقة لحرف B فى نهاية العبارة . فإذا عدلنا العبارة السابقة لتصبح :

```
10 LINE (10,30)- (70,100),1,BF
```

فإن الشكل الرباعى يصبح كالتالى :



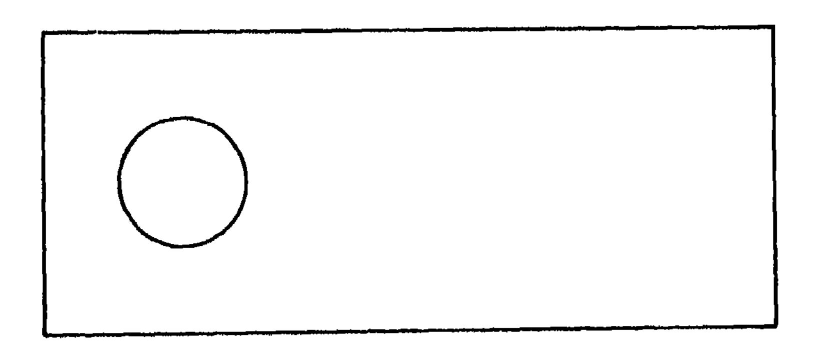
الدائرة : فى الرياضيات (علم الهندسة)، يمكننا رسم الدائرة إذا عرفنا (القطر أو نصف القطر) وإحداثيات مركز الدائرة، أما فى لغة البيسك فيجب معرفة طول نصف القطر وإحداثيات المركز، ومن ثم يمكننا رسم الدائرة باستخدام تعليمة دائرة CIRCLE، كما هو مبين بالشكل التالى :

الشكل العام لعبارة تعليمة دائرة				CIRCLE
N	, N,	(N1, N2)	CIRCLE	XXX
↓	↓	↓	↓	↓
لون خط الدائرة	طول نصف القطر	إحداثيات المركز	تعليمة دائرة	رقم السطر

فبالعبارة التالية :

```
10   CIRCLE (80, 80), 40, 1
```

نعمل على رسم دائرة إحداثيات مركزها (80,80) ونصف قطرها يساوى 40 ولون الخط الذى ستظهر به أزرق، كما هو مبين فيمايلى :



لاحظ وجود التعرجات فى الخط الدائرى ، و يعود ذلك إلى الحالة التى توجد بها الشاشة وهى متوسطة الجودة SCREEN 1 وإذا عملنا على تحويل الحالة إلى عالية الجودة SCREEN 2 فإن هذه التعرجات ستقل و يظهر الخط الدائرى متلاصقاً وأفضل بكثير مما كان عليه .

القوس : إن الزوايا تقاس بعدد درجاتها ، فمثلاً نقول إن الزاوية أ ب جـ تساوى ٩٠ درجة ، أى زاوية قائمة ، والزاوية المستقيمة تساوى ١٨٠ درجة ، والزاوية الكاملة تساوى ٣٦٠ درجة ، و يستخدم الباى PI (وهو الحرف السادس عشر من الأ بجدية اليونانية والذى يحمل القيمة العددية 3,1415 و يرمز له بالشكل π فى تمثيل النسبة مابين طول محيط الدائرة وقطرها .

ولرسم جزء من الدائرة نستخدم تعليمة دائرة ، كما هو مبين بالشكل التالى :

الشكل العام لعبارة تعليمة دائرة لرسم جزء من الدائرة						
N	N,	N,	N,	(N1, N2),	CIRCLE	XXX
نهاية الزاوية	بداية الزاوية	اللون	نصف القطر	إحداثيات مركز الدائرة	تعليمة دائرة	رقم السطر

فبالعبارة التالية :

```
10 CIRCLE (80,80), 40,1,.1,1.25
```

نعمل على رسم قوس أو جزء من الدائرة، كما هو مبين فيمايلى :

ولربط كل من بداية ونهاية القوس بمركز الدائرة نضع إشارة سالب أمام كل من بداية ونهاية الزاو ية .

فبالعبارة التالية :

```
10 CIRCLE (80,80), 40, 1,-.1,-1.25
```

نعمل على رسم جزء من الدائرة تتصل بمركزها، كما هو مبين فيمايلى :

أما العبارة التالية :

```
10 CIRCLE (80,80), 40,1,-PI,-3 *(PI /2)
```

فتعمل على رسم الربع الأ يسر السفلى من الدائرة، كما هو مبين فيمايلى :

تطوير برنامج باستخدام تعليمة الدائرة CIRCLE

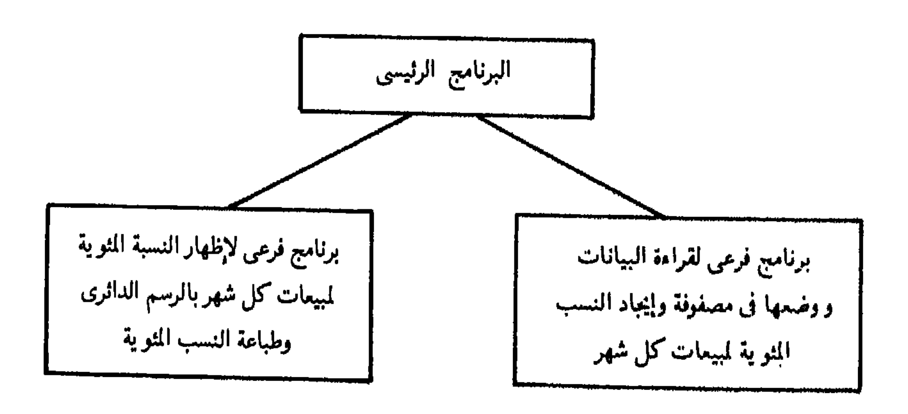
مثال (١٦ ــ ٣) : (الدائرة المقسمة)

الهدف : تمثيل مبيعات الشركة للاثنى عشر شهراً (مثال ١٦ ــ ١) بالرسم الدائرى المقسم (PIE CHART) .

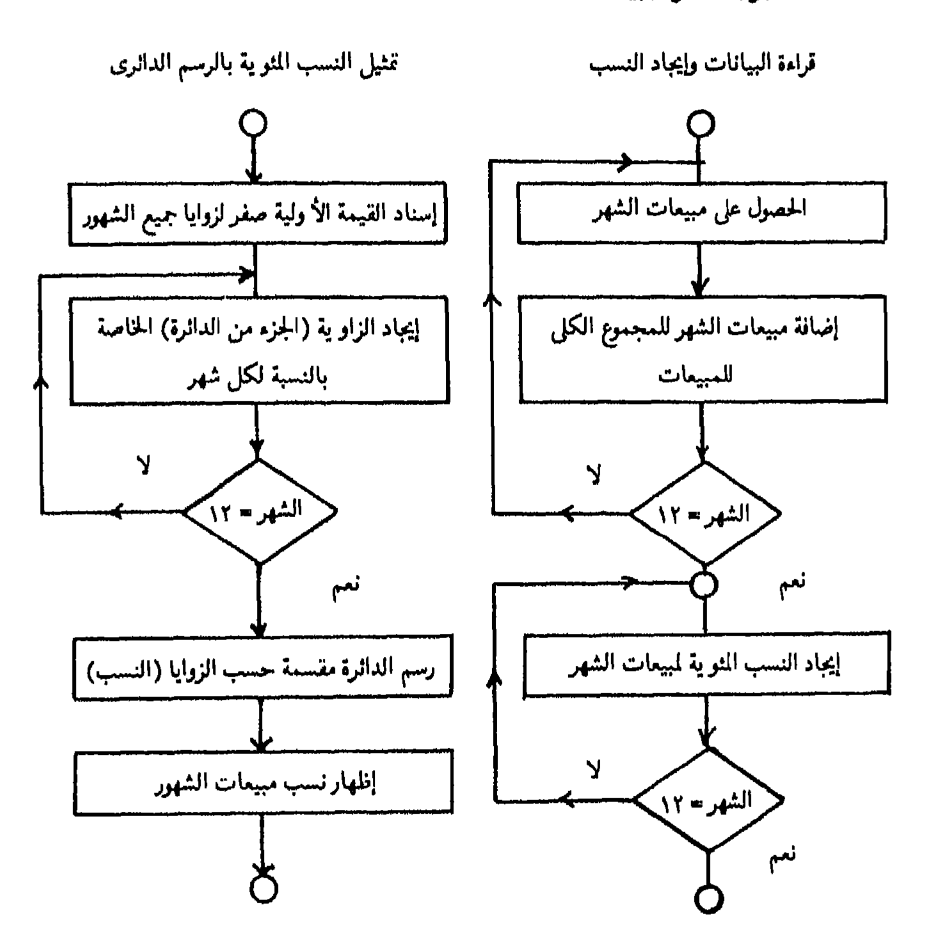
أولاً ــ خطوات الحل :

١ ــ الحصول على مبيعات الشهور الاثنى عشر ووضعها فى مصفوفة .
٢ ــ إيجاد المجموع الكلى لمبيعات الشركة خلال الاثنى عشر شهراً .
٣ ــ إيجاد النسبة المئوية لمبيعات كل شهر .
٤ ــ تمثيل النسبة المئوية لكل شهر حسب قيمتها فى الدائرة .
٥ ــ تكرار الخطوات ٣ و ٤ لجميع الشهور الاثنى عشر .
٦ ــ طباعة النسب المئوية لجميع الشهور.

ثانياً ــ الهيكل الهرمى :



ثالثاً ــ نمط البرمجة التركيبية :



رابعاً ــ البرنامج فى الشكل التالى :

شكل (١٦ ــ ٣)

برنامج لإظهار مبيعات الشركة للاثنى عشر شهراً برسم الدائرة المقسمة

```
10 CLS: REM  لمسح اي بيانات على الشاشه
20 SCREEN 2,1,0
30 DIM T(20),S(12),P(12),A(12)
40 GOSUB 70  :REM  أستدعاء برنامج فرعي لقرائة البيانات
50 GOSUB 210 :REM  أستدعاء برنامج فرعي لطباعة المخرجات
60 END
70 REM  برنامج فرعي
75 REM لقراءة البيانات ووضعها في مصفوفه وايجاد النسبه المئويه لمبيعات كل الشهر
80 LET T=0
90 FOR I =1 TO 12
100 READ S(I)
110 LET T = T + S(I) :REM  اضافة مبيعات الشهر لمجموع المبيعات
120 NEXT I
130 REM  للا يجاد النسبه المؤيه لمبيعات كل شهر
140 FOR I = 1 TO 12
150 LET P(I) = (S(I) / T)
160 NEXT I
170 REM  البيانات
180 DATA 6000,10000,13000,7500,10500,14000
190 DATA 9300,3500,10000,5000,11000,14000
```

```
200 RETURN
210 REM  برنامج فرعي للاظهار مبيعات كل شهر بالرسم الدائري
220 FOR I=0 TO 12 : REM  للاسناد القيمه الاوليه لزاوية كل شهر
230 LET A(I) = 0
240 NEXT I
250 REM  للايجاد الزاويه لمبيعات كل شهر
260 FOR I = 1 TO 12
270 LET D = P(I) * (2 * 3.141)
280 LET A(I) = A(I - 1 ) + D
290 CIRCLE (400,100),80,1,-A(I),-A(I-1)
300 NEXT I
310 REM  لطباعة النسبه المويه لمبيعات كل شهر
320 LOCATE 1,1:  PRINT " النسبه المؤية لمبيعاته    الشهر"
330 LOCATE 2,1:  PRINT "======================    ====="
340 FOR I=1 TO 12
350  PRINT USING      "         #.###                ## ";P(I),I
360 NEXT I
370 RETURN
```

وفى حالة تنفيذ البرنامج سنحصل على النتائج التالية المطبوعة :

الشهر	النسبة المئوية للمبيعات
1	0.053
2	0.088
3	0.114
4	0.066
5	0.092
6	0.123
7	0.082
8	0.031
9	0.088
10	0.044
11	0.097
12	0.123

أما الدائرة، فيمكن مشاهدتها على الشاشة .

تمارين

١ ـــ اعمل على طباعة وتنفيذ البرامج الموجودة فى أشكال الفصل، وإجراء التعديلات عليها إن وجدت، وتنفيذها حسب تسلسل هذه التعديلات .

٢ ـــ بين الأخطاء إن وجدت فى العبارات التالية :

أ
```
10   CIRCLE (20, 30)
```
ب
```
10   LENE (2,3) - (40,70),1,BN
```
جـ
```
10   LINE-STEP (70,-30)
```
د
```
10   SCREEN  7,40,20
```
هـ
```
10   PI = 4.314159
```

٣ ـــ اعمل على تعديل العبارات وإجراء البرامج التالية لكى يتم تنفيذها بدون أخطاء :

أ ـــ لرسم مستطيل
```
10  LINE (20,80) (70,120),1,B
```
ب ـــ لرسم قوس
```
10  CIRCLE (50,50) 20,1,.1,1.5
```
جـ ـــ لرسم خطوط توصل زوايا الشاشة الأربع
```
10  FOR I = 1 TO  80
20  LOCATE 5,I: PRINT CHR$ (196 )
30  NEXT I
40  FOR I = 1 TO 25
50  LOCATE 1,I: PRINT CHR$ (196);
60  NEXT I
```

٤ ـــ أكتب العبارات اللازمة لرسم كل مما يلى :

أ ـــ دائرة نصف قطرها ٣٠ وإحداثيات المركز (٦٠، ٥٠) .

ب ــ مثلث (أ ب جـ) مع العلم بأن إحداثيات أ = (٢٠، ٣٠)، ب = (٥٠، ٦٠)، جـ = (٩٠، ١٠٠) .

جـ ــ اعمل على تعديل كل من أ، ب بحيث تتم تغطية الأشكال من الداخل (تظليلها) .

٥ ــ فى التمرين رقم ٤ فى الفصل الثالث عشر، أظهر النتائج باستخدام النقاط المبعثرة SCATTERDIAGRAM

أ ــ رسم بالقيم الفعلية للأسهم .

ب ــ رسم لنسب التغيير .

٦ ــ يتسلم المركز الرئيسى لإدارة محلات بنده فى الرياض تقريراً فى نهاية كل يوم، فيه إجمالى المبيعات لكل فرع، يقوم محمد ــ السكرتير ــ كل أسبوع بتزويد المدير العام بتقرير فى صباح يوم السبت يحتوى على مجموعات المبيعات الأسبوعية لكل فرع من الفروع ونسبة مبيعات كل فرع إلى المجموع الكلى .

١ ــ طور برنامجاً للقيام بهذه المهمة علماً بأن هناك خمسة فروع لبندة فى الرياض .

٢ ــ أظهر النتائج بواسطة الدائرة المقسمة .

ملحق عمليات التشغيل

مقدمة عن عمليات التشغيل

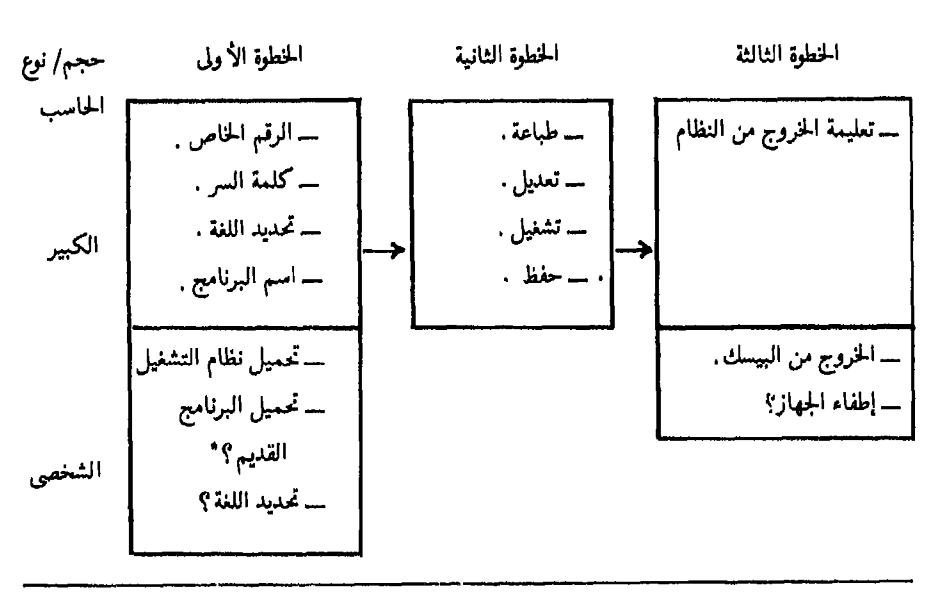
يمكن تقسيم الخطوات المتبعة فى التخاطب مع الحاسبات بأحجامها المختلفة إلى ثلاث خطوات رئيسية :

١) عملية الدخول إلى الحاسب وتعريف المستخدم واللغة .

٢) عملية التخاطب مع الحاسب مثل طباعة وتغيير البرامج وتشغيلها .

٣) عملية الانتهاء والخروج من الحاسب .

و يبين الرسم التالى تسلسل هذه الخطوات وأنواع الأنشطة والمعلومات المطلوبة لكل خطوة وذلك حسب حجم الحاسب .



* ؟ : إذا اقتضى الأمر ذلك .

لاحظ عدم وجود فروقات فى أنشطة الخطوة الثانية بين الحاسبات الآلية الكبيرة والصغيرة، والفروقات فى الخطوتين الأخريين ناتجة عن أن الحاسبات الكبيرة هى متعددة المستخدمين، وأن الحاسبات الشخصية ــ فى الغالب ــ هى ذات مستخدم واحد . وتعدد المستخدمين يؤدى إلى تعدد مقاصد الاستخدام ولغات التخاطب مع الحاسب .

ويعنى هذا أمرين :

١ ــ أن يوضع الحاسب فى حالة الاستعداد الدائم لقبول طلبات المستخدمين .
٢ ــ تعريف المستخدمين ونوع لغة التخاطب .

وسنبحث فى بقية الفصل الخطوات المحددة للتخاطب مع نوعين من الحاسبات الكبيرة IBM, HP-3000 وثلاثة أنواع من الحاسبات الشخصية :

آى . بى . أم IBM وراديوشاك RADIO-SHACK وأبل APPLE

خطوات التعامل مع الحاسبات الكبيرة

إن أسس خطوات التعامل مع الحاسبات الآلية الكبيرة MAIN FRAME COMPUTERS واحدة حتى مع اختلاف نوعية هذه الأجهزة، ففى جميع المؤسسات والشركات يوجد لكل مستخدم رقم خاص USER-ID وكلمة سر PASSWORD خاصة به، حيث يقوم كل مستخدم بإدخالهما ومن ثم يعمل نظام التشغيل باستدعاء برنامج خاص (مخزن به جميع الأرقام الخاصة بالمستخدمين وكلمة السر لكل رقم)؛ ليعمل على التأكد من صلاحية الرقم وكلمة السر للمحافظة على بيانات وأجهزة المركز، ففى حالة عدم تطابقهما يعمل الجهاز على إظهار رسالة بذلك، وفى معظم الحالات لايسمح لهذا المستخدم بالتعامل مع الجهاز .

وسنتعرض الآن لخطوات الدخول والتعامل مع بعض هذه الأجهزة :

أ) جهازآی بی إم I. B. M

١ ــ يتم التعامل مع معظم لغات البرمجة باستخدام نظام TSO وذلك بكتابة كلمة TSO ومن ثم الضغط على مفتاح العودة/ الإدخال ENTER .

٢ ــ سيسأل المستخدم عن إدخال رقمه بالرسالة التالية ENTER USERID .

٣ ــ بعد كتابة الرقم والضغط على مفتاح العودة/ الإدخال ENTER سيسأل المستخدم عن إدخال كلمة السر بالرسالة التالية

. ENTER CURRENT PASSWORD FOR USERID

٤ ــ بعد كتابة كلمة السر والضغط على مفتاح العودة/ الإدخال يتم التدقيق فى صلاحيتهما ، ففى حالة السماح للمستخدم باستعمال الجهاز ستظهر كلمة READY لإشعار المستخدم بأن الجهاز فى حالة الاستعداد لاستقبال التعليمات .

٥ ــ لكتابة برنامج بلغة بيسك تكتب الجملة التالية :

EDIT TEST (XXXXX) VSBASIC

↑

اسم البرنامج

ومن ثم الضغط على مفتاح العودة/ الإدخال .

ففى حالة كون البرنامج جديداً ستظهر كلمة INPUT وفى سطر آخر يظهر رقم 0010 للبدء فى كتابة جمل البرنامج ، وفى حالة الانتهاء من جميع جمل البرنامج والضغط على مفتاح العودة/ الإدخال ستظهر كلمة EDIT أى أن بإمكان المستخدم إجراء أى من العمليات المبينة فى جدول (ب) ، أما إذا كان البرنامج قد خزن سابقاً فسيتم تحميله تلقائياً .

٦ ــ وبعد الانتهاء من طباعة البرنامج أو تعديله يمكن حفظه بإحدى وسائل التخزين باستخدام العبارة التالية END SAVE ومن ثم الضغط على مفتاح العودة/ الإدخال ، بعد ذلك ستظهر كلمة READY .

٧ ــ للخروج من النظام نكتب تعليمة LOGOFF ومن ثم الضغط على مفتاح العودة/ الإدخال .

ب) جهاز هيوليت باكرد ٣٠٠٠ HEWLETT PACKARD 3000 SERIES III

١ ــ بالضغط على مفتاح العودة/ الإدخال يتم وضع الحاسب فى حالة الاستعداد لاستقبال التعليمات من المستخدم، وذلك بعد إظهار نقطتين «فوق بعض» : COLON .

٢ ــ بعد إظهار النقطتين تتم طباعة السطر التالى :

: HELLO XXXXX ; TERM = 6

رقم

ومن ثم الضغط على مفتاح العودة/ الإدخال .

٣ ــ بعد إدخال رقم المستخدم والتأكد من صحته سيطلب من المستخدم إدخال كلمة السر الخاصة به، و بعد إدخالها والضغط على مفتاح العودة/ الإدخال سيتم التأكد من صحتها أيضاً، وفى حالة صحتها سيتم إظهار عدة رسائل على الشاشة، بعدها يتم إظهار نقطتين «فوق بعض» (: COLON) لإشعار المستخدم بأن الجهاز فى حالة الاستعداد لاستقبال التعليمات .

٤ ــ للتعامل مع لغة بيسك تكتب كلمة BASIC ومن ثم الضغط على مفتاح العودة/ الإدخال . سيظهر بعد ذلك إشارة أكبر من > لإشعار المستخدم بأن الجهاز فى حالة الاستعداد لتلقى التعليمات الخاصة بلغة بيسك .

٥ ــ لإدخال برنامج جديد نبدأ بكتابة العبارات حسب قواعد وتعليمات لغة بيسك والضغط على مفتاح العودة/ الإدخال بعد الانتهاء من كل عبارة . أما لاستدعاء برنامج قديم فنستخدم العبارة اسم البرنامج GET XXX .

٦ ــ فى حالة الانتهاء من كتابة/ تعديل/ تنفيذ البرنامج يمكن حفظه بإحدى وسائل التخزين بكتابة تعليمة احفظ واسم البرنامج كالتالى: اسم البرنامج XXX

SAVE ومن ثم الضغط على مفتاح العودة/ الإدخال . و بعد ذلك ستظهر إشارة أكبر من > أى أن بإمكان المستخدم التعامل مع أى برنامج من برامج لغة بيسك بإجراء أى من العمليات المبينة فى جدول (أ) .

٧ ـ للخروج من نظام بيسك إلى نظام التشغيل نكتب تعليمة نظام SYSTEM ومن ثم الضغط على مفتاح العودة/ الإدخال ، ستظهر بعد ذلك نقطتان «فوق بعض» (:) ولإنهاء التعامل مع الجهاز نكتب تعليمة BYE ومن ثم الضغط على مفتاح العودة/ الإدخال .

خطوات التعامل مع الحاسبات الآلية الشخصية

إن خطوات التعامل مع الحاسبات الآلية الشخصية PERSONAL COMPUTERS تختلف إلى حد ما عن التعامل مع الحاسبات الآلية الكبيرة ؛ ففى الحاسبات الكبيرة يعمل المشغل THE OPERATOR على تجهيز الحاسب ووضعه فى حالة الاستعداد للمستخدمين وذلك بتحميل نظام التشغيل للذاكرة الرئيسية ، أما فى حالة الحاسبات الشخصية ، والتى يطلق عليها فى بعض الأحيان اسم الحاسبات الصغيرة SMALL COMPUTERS ، فيقوم المستخدم بتجهيز الجهاز وتحميل نظام التشغيل للذاكرة الرئيسية باتباع الخطوات التالية :

أ) وضع الأسطوانة التى تحتوى على نظام التشغيل و يطلق عليها اسم DISK OPERATING SYSTEM (DOS) فى وحدة الأسطوانة الرئيسية DISK DRIVE والتى عادة ما تحمل الرقم 1 أو الحرف A .

ب) توصيل التيار الكهربائى وذلك بوضع مفتاح الكهرباء فى حالة ON .

وسنتعرض الآن لكيفية التعامل مع لغة بيسك باستخدام ثلاث حاسبات آلية شخصية مختلفة :

أولاً ـــ جهاز آى بى إم I.B.M

١ ـــ بعد تنفيذ الخطوتين (أ، ب) سيبدأ البحث عن نظام التشغيل DOS ومن ثم تحميله للذاكرة الرئيسية . بعد ذلك سيطلب من المستخدم إدخال التاريخ والوقت .

ففى حالة إدخال التاريخ حسب الصيغة المظهرة سيحل فى الذاكرة الرئيسية مكان التاريخ الأصلى، ويمكننا الضغط على مفتاح العودة وإبقاء التاريخ القديم دون أن يؤثر على سير العمل .

٢ ـــ بعد الانتهاء من إدخال التاريخ والوقت سيظهر حرف A متبوعاً بإشارة أكبر من (A >) وهذا يعنى أن الجهاز فى حالة الاستعداد لاستقبال أى تعلميات والتعامل مع الأسطوانة الموجودة فى الوحدة A .

٣ ـــ للتعامل مع نظام البرمجة بلغة بيسك يمكننا استدعاء نظام BASIC أو نظام * BASICA (وهو عبارة عن نظام BASIC المعدل بإضافة بعض التعليمات المتقدمة) . و يتم ذلك بكتابة مترجم اللغة المتوفر ومن ثم الضغط على مفتاح العودة/ الإدخال (↵) .

٤ ـــ بعد الاستدعاء سيتم البحث عن النظام وتحميله للذاكرة الرئيسية و بعد الانتهاء من ذلك ستظهر الرسائل التالية :

```
THE IBM PERSONAL COMPUTER BASIC
VERSION 2.1 COPYRIGHT  IBM., CORP., 1981, 1982, 1983
 61327 BYTES FREE
```

* سيتم شرح خطوات التعامل مع BASICA فى الجزء الأخير من هذا الفصل.

٥ ــ هنا يكون الجهاز فى حالة الاستعداد للتعامل مع لغة بيسك . فإذا أردنا إدخال برنامج نبدأ بكتابة العبارات والضغط على مفتاح العودة/ الإدخال بعد الانتهاء من كل عبارة إلى أن ننتهى من كتابة البرنامج .

٦ ــ يمكننا التعامل مع هذا البرنامج باستخدام أى من العمليات المبينة فى جدول (أ) .

ثانياً ــ جهاز أبل APPLE

١ ــ بعد تنفيذ كل من الخطوتين (أ، ب) سيبدأ البحث عن نظام التشغيل DOS ومن ثم تحميله للذاكرة الرئيسية، وبعد الانتهاء ستظهر الرسائل التالية على الشاشة :

DOS VERSION # # * # # / # # / # # **

APPLE II PLUS OR ROMCARD SYSTEM MASTER

(LOADING INTEGER INTO LANGUAGE CARD)

٢ ــ بعد ظهور المؤشر CURSOR لإشعار المستخدم بأن الجهاز فى حالة الاستعداد لاستقبال التعليمات، يجب كتابة كلمة جديد NEW لحذف أى برنامج يتعلق بلغة بيسك ومحمل بالذاكرة الرئيسية .

٣ ــ بعد الانتهاء من كتابة NEW والضغط على مفتاح العودة سيصبح الجهاز فى حالة الاستعداد لاستقبال أى تعليمة تتعلق بلغة بيسك .

٤ ــ إذا أردنا إدخال برنامج جديد نبدأ بكتابة العبارات، ومن ثم الضغط على مفتاح العودة بعد الانتهاء من كل عبارة إلى أن ننتهى من كتابة البرنامج .

٥ ــ بعد ذلك يمكننا التعامل مع هذا البرنامج باستخدام أى من التعليمات المبينة فى جدول (أ) .

---

* رقم نسخة نظام التشغيل .

** التاريخ .

ثالثاً ــ جهاز راديو شاك RADIO SHACK

١ ــ بعد تنفيذ كل من الخطوتين (أ، ب) سيبدأ البحث عن نظام التشغيل DOS ومن ثم تحميله للذاكرة الرئيسية .

٢ ــ بعد الانتهاء من مرحلة التحميل سيظهر المؤشر على الزاوية اليسرى العلوية من الشاشة، وعندها يكون الجهاز فى حالة الاستعداد لتلقى التعليمات من المستخدم .

٣ ــ للتعامل مع لغة بيسك ندخل التعليمة BASIC ، ومن ثم الضغط على مفتاح العودة/ الإدخال، بعد ذلك ستظهر الرسائل التالية :

```
RADIO SHACK MODEL III BASIC
(C)' 80 Tandy
READY
 >
```

٤ ــ هنا يكون الجهاز فى حالة الاستعداد للتعامل مع لغة بيسك، فإذا أردنا إدخال برنامج نبدأ بكتابة العبارات والضغط على مفتاح العودة/ الإدخال بعد الانتهاء من كل عبارة إلى أن ننتهى من كتابة البرنامج .

٥ ــ يمكننا التعامل مع هذا البرنامج باستخدام أى من التعليمات المبينة فى جدول (أ) .

جدول (١٧ ـ أ)

بعض العمليات والتعليمات المتعلقة بكل منها في الحاسبات الآلية الشخصية

نوع العملية	جهاز آي بي إم I.B.M	جهاز أبل APPLE	جهاز راديوشاك RADIO SHACK
١ ـ لإظهار البرنامج على الشاشة	LIST	LIST (RETURN)	LIST (ENTER)
٢ ـ لتنفيذ البرنامج.	RUN	RUN «اسم البرنامج»	RUN
٣ ـ حفظ البرنامج.	SAVE «اسم البرنامج»	SAVE «اسم البرنامج»	SAVE «اسم البرنامج»
٤ ـ لتحميل البرنامج.	LOAD «اسم البرنامج»	LOAD «اسم البرنامج»	LOAD «اسم البرنامج»
٥ ـ للحصول على نسخة مطبوعة من البرنامج.	LLIST	PR#0 LIST	LLIST
٦ ـ لطباعة النتائج.	LPRINT	PR#1 PRINT	LPRINT
٧ ـ لإظهار جزء من البرنامج على الشاشة.	LIST إلى ـ من	LIST إلى ـ من	LIST إلى ـ من
٨ ـ لطباعة جزء من البرنامج.	LLIST إلى ـ من	PR#0 LIST إلى ـ من	LLIST إلى ـ من
٩ ـ مسح الشاشة وإعادة المؤشر للزاوية اليسرى العلوية.	GLS	HOME	GLS
١٠ ـ لحذف عبارة من البرنامج.	طباعة رقم العبارة	طباعة رقم العبارة	طباعة رقم العبارة
١١ ـ لحذف أكثر من عبارة من البرنامج.	DELETE إلى ـ من	DELETE إلى ـ من	DELETE إلى ـ من
١٢ ـ لحذف البرنامج من وحدة التخزين.	KILL اسم البرنامج	DELETE «اسم البرنامج»	DELETE «اسم البرنامج»
١٣ ـ للعودة من نظام بيسك لنظام التشغيل.	SYSTEM	إعادة تجهيز الجهاز بالضغط على مفتاح ESC	TRSDOS

جدول (١٧ ـــ ب)

بعض العمليات والتعليمات المتعلقة بكل منها فى الحاسبات الآلية الكبيرة		
نوع العملية	**جهاز آى بى إم I. B. M.**	**HP - 3000 SERIES III**
١ ـــ إظهار البرنامج على الشاشة .	LIST	LIST
٢ ـــ تنفيذ البرنامج .	RUN	RUN
٣ ـــ حفظ البرنامج .	END SAVE	SAVE
٤ ـــ لتحميل البرنامج .	EDIT TEST (اسم البرنامج) VS BASIC	GET اسم البرنامج
٥ ـــ حذف عبارة من البرنامج .	طباعة رقم العبارة	طباعة رقم العبارة
٦ ـــ حذف أكثر من عبارة من البرنامج .	DELETE إلى رقم عبارة ـــ من رقم عبارة	DEL إلى رقم عبارة ـــ من رقم عبارة
٧ ـــ حذف البرنامج من وحدة التخزين .	DELETE X «رقم المستخدم» TEST. VSBASIC (اسم البرنامج)	PURGE اسم البرنامج
٨ ـــ تغيير اسم البرنامج .	RENAME رقم المستخدم TEST . VSBASIC (اسم البرنامج) (الاسم الجديد) TEST. VSBASSC . رقم المستخدم	اسم البرنامج الجديد واسم البرنامج القديم RENAME
٩ ـــ للحصول على نسخة مطبوعة من البرنامج.	DSPRINT ، رقم المستخدم TEST. VSBASIC (اسم البرنامج) اسم الطابعة	: FILE PRINT; DEV = LP : BASIC > GET اسم البرنامج > LIST, OUT = PRINTER
١٠ ـــ للحصول على النتائج مطبوعة فى الحاسبات الآلية الكبيرة وخاصة آى بى إم نحتاج إلى لغة التحكم فى التشغيل JCL وهى اختصار لـ Job Control Card	// STEPOL EXEC PGM = ICDOS BSC // SYSPRINT DD SYSOUT =* // CONTROL DD* RUN ABC SOURCE // * // ABC DD* [البرنامج] // * //	: FILE PRINTER; DEV = LP : BASIC > GET اسم البرنامج > RUN, OUT = PRINTER

الإجراءات الخاصة بطباعة البرامج المطورة فى الكتاب

تـم تـطـويـر جميع البرامج باستخدام جهاز آى بى إم IBM الشخصى العادى، سعة الـذاكـرة ٢٥٦ ألـف حرف (256 K) واستخدام لغة BASICA فى كتابة البرامج والتى تسمح بالتعامل مع البيانات فى اللغتين العربية والإنجليزية .

وللتـعـامل مع BASICA لابد من توفر النظم المتضمنة فى A S D والتى تكون مخزنة على قرص نظام التشغيل وتتبع الخطوات التالية فى عملية التشغيل :

١ ــ تحميل قرص نظام التشغيل والذى يحتوى على البرامج التالية :

أ) WTDATIM و يتم تحميله تلقائياً بعد تشغيل الجهاز .

ب) ARABIC و يـتـم إدخـالهـا مـن قـبـل المـسـتـخدم ، ومن ثم الضغط على مفتاح الإدخـال/الـعـودة لـتـحـمـيـل الحـروف والأ رقـام والـرمـوز العربية فى الذاكرة الرئيسية ، وبعد إتمام ذلك ستظهر الرسالة التالية :

ASD OPTION SPECIFIED AND LOADED

جـ) AMODE A و يـتـم إدخـالهـا مـن قـبـل المـسـتـخدم ومن ثم الضغط على مفتاح الإدخـال/الـعـودة لـتعريف الحروف والأ رقام والرموز العربية المظهرة على لوحة المفاتيح .

٢ ــ طباعة BASICA لتحميل مترجم لغة بيسك .

٣ ــ يتم طباعة تعليمات بيسك باللغة الانجليزية .

٤ ــ للتـحـويـل مـن حـالـة اللـغة الإنجليزية إلى حالة اللغة العربية ، من أجل طباعة العناوين أو الرسائل أو البيانات أو الشرح والتعليق ، تتبع الخطوات التالية :

أ) الـضـغـط المـستمر على مفتاح (+ ALT) ومن ثم الضغط على المفتاحين التاليين

بالتوالى (↑) الموجودين فى الجهة اليمنى من لوحة المفاتيح ومفتاح (⇆) الموجود فى الجهة العلوية اليسرى من لوحة المفاتيح .

وفى حالة إتمام هذه العمليات بنجاح يصدر الحاسب صفيراً قصيراً . وفى حالة عدم سماع الصفير تتم إعادة هذه الخطوة .

ب) للعودة إلى حالة اللغة الإنجليزية يتم الضغط المستمر على مفتاح (ALT) ومن ثم الضغط على مفتاح (↑) الموجود فى الجهة اليسرى من لوحة المفاتيح .

مقارنة لغة بيسك بلغتى البرمجة شائعتى الاستخدام
(فورتران ، كوبول)

ستكون أسس المقارنة وفق العوامل الأساسية التالية :

أولا — الطبيعة العامة للغة :

تحديد الهدف الأساسى من تطوير اللغة ، طبيعتها العامة وذلك كالتالى :

أ — لغة بيسك : تستخدم لغة بيسك لتسهيل عملية التخاطب المباشر مع الحاسبات الشخصية والتى استخدمت بشكل خاص فى المؤسسات التعليمية ، وكلمة بيسك BASIC باللغة الإنجليزية هى الأحرف الأولى من الكلمات التالية :

Beginners All-purpose Symbolic Instruction Code.

ب — لغة فورتران : تلائم لغة فورتران توصيف العمليات الحسابية المعقدة كتلك التى فى التطبيقات العلمية المجردة ، وكلمة فورتران FORTRAN باللغة الإنجليزية مقتبسة من الكلمتين

FORMALA TRANSLATOR

جـ — لغة كوبول : تناسب لغة كوبول التطبيقات التجارية والتى تتطلب إنتاج تقارير دون الاعتماد على عمليات حسابية معقدة ، وقد اشتقت كلمة كوبول COBOL من الكلمات التالية :

Common Business Oriented Language

ثانياً — تركيبة البرنامج :

بناء على الطبيعة العامة للغة نجد أن درجة تركيبة البرنامج تتفاوت مابين اللغات الثلاث ، فلغة بيسك تعتبر من أكثر اللغات سهولة حيث لا تتبع تركيباً معيناً ، أما لغة

فورتران فلها بعض القوانين المحددة و بشكل خاص تلك التى توضح كيفية إدخال البيانات وإظهارها ، أما لغة كوبول فهى أكثر اللغات تقيداً بقوانين محددة لأجزاء البرنامج ، حيث إن كل برنامج يحتوى على أجزاء معينة بغض النظر عن الهدف منه .

ثالثاً ــ أسس البرمجة الهيكلية والتركيبية :

ذكرنا فى المقدمة عاملين أساسيين يسهل توفرهما اتباع أساليب التجزئة الهرمية والبرمجة التركيبية :

أ) تحديد بدايات ونهايات الأجزاء فى البرنامج الواحد .

ب) التحكم فى تكرار تنفيذ أجزاء البرنامج .

لايوجد هناك اختلاف كبير بين اللغات الثلاث فى مدى قابليتها لتوفير العاملين المذكورين آنفاً ، ولكن قد نجد أن اتباع الأساليب الحديثة فى البرمجة ميسر فى لغة كوبول أكثر منها فى اللغتين الأخريين.

رابعاً ــ سهولة فهم البرامج المكتوبة باللغات :

نجد أن لغة كوبول هى أقرب لغات البرمجة إلى اللغة الإنجليزية العادية ، لذلك نرى ضرورة تناثر العبارات التوضيحية فى لغتى بيسك وفورتران .

وفيما يلى جدول تفصيلى يبين المقارنة بين اللغات الثلاث وفق بعض العمليات:

جدول (١٧ — جـ) مقارنة بين التعليمات والقواعد الخاصة بها فى لغات البرمجة (بيسك، فورتران وكوبول)

نوع العملية	لغة بيسك	لغة فورتران	لغة كوبول
أسماء المتغيرات VARIABLE NAMES	حرف أو حرف متبوع برقم A, A0, A1/ ... A9 أو ٣٠ حرفاً فى بعض الحاسبات	من حرف إلى ستة أحرف C, COUNT, COUNT 4	من حرف إلى ٣٠ حرفاً READ, PRINT - ROUTINE
تصنيف البيانات واختيار أسماء المتغيرات لها وتوصيفها DATA TYPES	قيم عددية = N قيم غير عددية = N$	— قيم عددية صحيحة = يجب أن يبدأ اسم المتغير بأى من الحروف التالية I,J,K,L,M,N — قيم عددية غير صحيحة وغير عددية أى من الحروف A—H و O—Z	— قيم عددية صحيحة PIC 999 — قيم عددية غير صحيحة PIC 9V99 — قيم غير عددية PIC XXX
عناوين العبارات وأجزاء البرنامج LABELS	رقم سطر...10 20....	رقم 10	اسم يطلق على الجزء READ - ROUTINE.
أعمدة ذات استخدام خاص SPECIAL COLUMNS	لا يوجد	من ١ — ٥ لعناوين العبارات ٦ للتكملة من السطر السابق	من ٨ — ٣٧ أسماء الأجزاء الرئيسية وأسماء العبارات. ١٢ — ٧٢ العبارات. ٧ للتكملة أو الشرح.
للشرح والتعليق على العمليات REMRKS / COMMENTS	نستخدم تعليمة REM	يستخدم الحرف C فى العمود 1	يستخدم الرمز ٭ فى العمود ٧
آخر تعليمة فى البرنامج — لإنهاء العمليات فى البرنامج والتوقف. LAST STATEMENT	END END	END STOP	لم يحدد STOP - RUN

MOVE 8 TO A.	A = 8	LET A = 8	إسناد قيمة للمتغير MOVE/ASSIGN
READ DATA-REC-IN ATEND MOVE « NO » TO FLAG-EOD. IF FLAG-EOD =«NO»PERFORM 500-END CLOSE FILE-IN, FILE-OUT.	READ (5,10)A,N 10 FORMAT (F3.1,I2)	READ A,N DATA 5,14	قراءة البيانات READING DATA
WRITE DATA-REC-OUT AFTER ADVANCING 2 LINES.	WRITE (6,20) A,N 20 FORMAT (10X,F4.1,I4)	PRINT A,N	طباعة نتائج المعالجة PRINTING RESULTS
ADD B TO C GIVING A.	A = B + C	LET A = B + C	إجراء عمليات حسابية ARITHMETIC OPERATIONS
A OCCURS 10 TIMES PIC 99.	DIMENSION A(10), B (5,20)	DIM A (10), B(5,20)	حجز أماكن للقيم فى الذاكرة ARRAYS (مصفوفات)
PERFORM 100-READ-AND-PROCESS 6 TIMES.	DO 20 I = 1,6 ⋮ 20 CONTINUE	FOR I = 1 TO 6 ⋮ NEXT I	الدوارة والتحكم فى تنفيذ الأجزاء LOOPS
IF A = 6 PERFORM 300-PRINT.	IF (A.EQ.6) QOTO 20	IF A = 6 THEN 300	إجراء عملية مقارنة/ منطقية والتشعب COMPARISONS
DISPLAY « Enter the Value: »	WRITE (6,20) 20 FORMAT(10X; EntertheValue:')	PRINT « Enter the Value : »	لطباعة الرسائل (التخاطب مع المستخدم) MESSAGES INTERACTIVE

المراجع العربية

أ ــ معالجة البيانات :

ــ عوض منصور، محمد أبو النور، محمد العمرى : «**مقدمة فى علم الحاسب الإلكترونى**»، الأردن، ١٩٨٧ .

ــ محمد الفيومى : «**مقدمة فى علم الحاسبات الإلكترونية ومعالجة البيانات**»، دار الفرقان، الأردن، ١٩٨٤ .

ــ محمد شوقى بشارة : «**الحاسبات الإلكترونية ونظم المعلومات**»، بيروت، ١٩٨٣ .

ب ــ البرمجة بلغة البيسك :

ــ عوض منصور: «**برمجة بيسك للمبتدئين**» الأردن، ١٩٨٧م.

ــ مظهر طايل : «**الكمبيوتر لغة وأداء (بيسك)**»، بيروت، ١٩٨٤ .

ــ محمد الفيومى : «**برمجة الحاسبات الإلكترونية بلغة بيسك**»، دار الأمل، الأردن، ١٩٨٤ .

ــ محمد السعيد خشبة : «**أساليب تخطيط البرامج بلغة البيسك**»، جمهورية مصر العربية، ١٩٨٤ .

المراجع الأجنبية

أ ــ معالجة البيانات :

— Basili. Victor R. and Baker, F. Terry, "**Tutorial on Structured Programming and Intigrated Practices,**" Los Almitos CA, IEEE Computer Science Press, 1981.

— Bohl, Marilyn, "**Flowcharting Techniques,**" Chicago, Science Research Associates, 1971.

— Bohl, Marilyn, "**Tools for Structured Design,**" Chicago, Science Research Associates, 1978.

— Bohl, Marilyn, "**Essentials of Information Processing,**" Chicago, Science Research Associates, 1986.

- Buzy, Beth Moorer, **"Using Computers,"** Chicago, Science Research Associates, 1985.

- Chapin, N., **"Flowcharts,"** Princeton, Auerbach Publishers, 1971.

- Charette, Robert N., and Stockenberg, John, **"A Unified Methodolgy for Developing Systems,"** New York, McGrew-Hill Book company, 1986.

- Dahl, Olew Johan, and Others, **"Structured Programming,"** New York, Academic Press, 1972.

- Eliason, Alan L., **"Business Information Processing,"** Chicago, Science Research Associates, 1979.

- Essick, Edward L., **"Principles of Business Data Processing,"** 3rd ed., Chicago, Science Research Associates, 1986.

- Farina, Mario V., **"Flowcharting,"** Englewood Cliffs, Prentice-Hall, 1970.

- Gear, C. William, **"Introduction to Computers, Structured Programming, and Applications,"** Chicago, Science Research Associates, 1978.

- Gear, C. William, **"Computer Applications and Algorithms,"** Chicago, Science Research Associates, 1986.

- Keringham, Brian W., and Plauger, P. J., **"Elements of Programming Style,"** New York, McGrew-Hill Book Company, 1978.

- LaBudde, Keith P., **"Structured Programming Concepts,"** New York, McGrew-Hill Book Company, 1987.

- Lebtz, B. P., Swanson E. B., and Tompkins, G. E, **"Characteristics of Application Software Maintenance,"** Communication of the ACM, Vol. 21, No. 6 (June 1987).

- Leeson, Marjorie M., **"Programming Logic,"** Chicago, Science Research Associates, 1983.

- Leeson, Marjorie M., **"Computer Operations,"** 3rd., Chicago, Science Research Associates, 1987.

- Leeson, Marjorie M., **"Computer Information : A Modular System,"** Chicago, Science Research Associates, 1985.

- Linger, Richard C., and Others, **"Structured Programming : Theory, and Practice,"** Reading Mass., Addison-Wesley, 1979.
- Mader, Chris, **"Information Systems,"** Chicago, Science Research Associates, 1979.
- Martin, James, and McClure, Carma L., **"Structured Techniques for Computing,"** Englewood Cliffs, Prentice-Hall, 1985.
- Maurer, Ward D., **"Programming : An Introduction to Computer Techniques,"** San Francisco, Holden-Day, 1972.
- Maynard, Jeff, **"Modular Programming,"** Princeton, Auerbach Publisher, 1972.
- McGrower, Clement L., and Kelly, John R., **"Top-Down Structured Programming Techniques,"** New York, Petrocelli/Charter, 1975.
- O'Leary, T. J., and Williams, B. K., **"Computers and Information Processing,"** Menlo Park CA, Benjamin/Cummings, 1985.
- Radlow, James, **"Computers and the Information Society,"** New York, McGrew-Hill Book Company, 1986.
- Ralston, Anthony, **"Introduction to Programming and Computer Science,"** New York, McGrew-Hill Book Company, 1971.
- Rothman, Stanley, and Mosmann, Charles, **"Computer Uses and Issues,"** Chicago, Science Research Associates, 1985.
- Sanders, Donald H., **"Computer Concepts and Applications,"** New York, McGrew-Hill Book Company, 1987.
- Sanders, Donald H., **"Computers Today,"** New York, McGrew-Hill Book Company, 1985.
- Shelly, Gary B., and Cashman, Thomas J., **"Computer Fundamentals with Application software,"** Boston, Boyd & Frazer Publishing Company, 1986.
- Spencer, Donald D., **"Computers and Information Processing,"** Columbus, Merrill Publishing Company, 1985.
  Spencer, Donald D., **"Computers : An Introduction ,"** Columbus, Merrill Publishing Company, 1986.

- Topping, Anne L., and Gibbons, Ian, **"Programming Logic : Structured Design,"** Chicago, Science Research Associates, 1985.
- Welsh, James, and McKeag, R. M., **"Structured System Programming,"** Englewood Cliffs, Prentice-Hall, 1980.
- Wetzel, Gregory F., and Bulgren, William G., **"The Algorithmic Process : An Introduction to Problem Solving,"** Chicago, Science Research Associates, 1985.
- Edward Yourdan, **"Techniques of Program Structure and Design,"** Englewood Cliffs, Prentice-Hall, Inc., 1975,
- Yourdon, Edward, **"Managing the Structured Techniques,"** 3rd ed., New York, McGrew-Hill Book Company, 1986.

ب ـ البرمجة بلغة بيسك :

- Alonso, J. R. F., **"Simple, BASIC Programs for Business Applications,"** Englewood Cliffs, Prentice-Hall, 1981.
- Benton, Stan, and Weeles, Len, **"Program It Right : Structured Methods in BASIC,"** Englewood Cliffs, Prentice-Hall, 1985.
- Bogart, Theodore F., **"Applied BASIC for Technology,"** Chicago, Science Research Associates, 1984.
- Boggs, Roy A., **"Applied BASIC for Microcomputer,"** Reston, VA, Reston Publishing Co., 1984.
- Boillot, Michel H., **"BASIC Concepts and Structured Problem Solving,"** St. Paul, West Publishing Co., 1984.
- Bosworth, Bruce, **"Business Programming Projects with BASIC,"** Chicago, Science Research Associates, 1984.
- Bradley, Julia C., **"MICROSOFT BASIC Using Modular Structure,"** Dobuque, Iowa, Wm. C. Brown Publishers, 1988.
- Chays Ruth K. and Miller, Joan M., **"More BASIC Programming for the Classroom and Home Teacher,"** New York, Teachers College Press, 1985.
- Compusoft, and Lien, David A. **"BASIC Programming for the IBM Personal Computer,"** Dubuque, Iowa Wm. C. Brown Publishers, 1984.

— Cox, Michael J., and Sullivan, Kathleen B., **"Structuring Programs in MICROSOFT BASIC,"** Boston, Boyd & Frazer Publishing Company, 1987.

— Dock, V. Thomas, **"BASIC Programming for Business,"** St. Paul, West Publishing Co., 1977.

— Ettlin, Walter A., and Solberg, Gregory, **"The MICROSOFT BASIC Book : Macintoch Edition,"** New York, McGrew-Hill Book Company, 1985.

— Finkel, LeRoy, and Brown, Jerald, **"Data File Programming in BASIC,"** New York, John Wiley and Sons, 1981.

— Goldstein, Larry J., **"IBM PC : An Introduction to the Operating System, BASIC Programming," 3rd ed., Bowie, MD, R. J. Brady Co., 1986.**

— **Gorsl, Jerome R., and Hills, William F., "Debugging Techniques for IBM PC BASIC,"** New York, Brady Communication Co., 1986.

— Grame, Carl A., and O'Donnell, Daniel J, **"Learning BASIC,"** Chicago, Science Research Associates, 1984.

— Grout, Jarrell C., **"Programming with BASIC : A Structured Approach,"** Dubuque, Iowa Wm. C. Brown Publishers, 1985.

— Hearn, Donald, and Baker, M. Pauline, **"Computer Graphics for the IBM Personal Computer,"** Englewood Cliffs, Prentice-Hall, 1983.

— Hergert, Douglas, **"MICROSOFT Quick BASIC : Developing Structure Programs,"** Redmond, Wash., Microsoft Press, 1987.

— High-Speed Sorting Algorithm, "Communications of the ACM, July 1959, Vol. 2, PP. 30-32.

— Hirschfelder, R., and Others, **"Structure Vax BASIC,"** Menlo Park CA, Benjamin/Cummings,1987.

— IBM Personal Computer Hardware Reference Library BASIC, P. 39.

— James, Mike, **"Artificial Intelligence in BASIC,"** Boston, Butlerworth, 1984.

— Keogh, James E., **"Programmer's Notebook : Modular Programming for Home Computers,"** New York, Simon and Schuster, 1984.

- Kittner, M., and Northcutt, B., "**BASIC : A Structured Approach,**" Menlo Park CA, Benjamin/Cummings, 1987.
- Kittner, M., and Norhcutt. B., "**Basic BASIC : A Structured Approach,**" 2nd ed., Menlo Park CA, Benjamin/Cummings, 1987.
- Ledgard, Henry, and Singer, Andrew, "**Elementary BASIC,**" Chicago, Science Research Associates, 1982.
- Lesser, Murray L., "**Using the MICROSOFT Business BASIC Compiler on the IBM PC,**" New York, McGrew-Hill Book Company, 1986.
- Lien, David A., "**BASIC Handbook : Encyclopedia of the BASIC Computer Language,**" 3rd ed., San Diago, Compusoft, 1986.
- Orilla, Lawrence S., "**Structured BASIC : An Integrated Approach,**" New York, McGrew-Hill Book Company, 1985.
- Peckham, Herbert, "**Hands-on BASIC for the Apple II Plus Computer,**" New York, McGrew-Hill Book Company, 1984.
- Peckham, Herbert, "**Hands-on BASIC for the DEC Professional,**" New York, McGrew-Hill Book Company, 1985.
- PecKham, Herbert, and Others, "**Structured BASIC for the IBM PC : A Hands-on Approach,**" New York, McGrew-Hill Book Company, 1985.
- Peckham, Herbert, "**Hands-on BASIC for the Apple IBM Personal Computer,**" New York, McGrew-Hill Book Company, 1983.
- Peckham, Herbert D., "**Hands-on BASIC for the Commodore 64,**" New York, McGrew-Hill Book Company, 1984.
- Peckham, Herbert D., "**Intermediate BASIC for the TI Home Computers,**" Englewood Cliffs, McGraw-Hill, Book Company, 1979.
- Peckham, Herbert D., "**Hands-on BASIC for the TRS-BO Color,**" Englewood Cliffs, McGraw-Hill, Book Company, 1983.
- Peckham, Herbert D., "**BASIC : A hands-on Method,**" 2nd ed., Englewood Cliffs, McGraw-Hill, Book Company, 1981.

- Peckham, Herbert D., **"MS DOS Structured BASIC,"** Englewood Cliffs, McGraw-Hill, Book Company, 1985.

- Quasney, James S, and Maniotes, John, **"Applesoft BASIC Fundamentals and Style,"** Boston, Boyd & Frazer Pubishing Company, 1987.

- Quasney, James S, and Maniotes, John, **"BASIC Fundamentals and Style,"** Boston, Boyd & Frazer Pubishing Company, 1987.

- Quasney, James S, and Maniotes, John, **"Complete BASIC for A Short Course,"** Boston, Boyd & Frazer Pubishing Company, 1987.

- Quasney, James S, and Maniotes, John, **"Structured BASIC Fundamentals and Style for the IBM PC and Compatibles,"** Boston, Boyd & Frazer Pubishing Company, 1988.

- Ruder, Jesse H., and Millsap, Gary, **"BASIC fo the IBM PC,"** New York, John Wiley and Sons, 1987.

- Ruder, Jesse H., **"BASIC fo the HP 3000,"** New York, John Wiley and Sons, 1986.

- Sawatzky, Jasper, and Chen, Shu-jen, **"Programming In BASIC PLUS,"** 2nd ed., New York, John Wiley and Sons, 1985.

- Scheinder, David I., **"Programming and Its Applications : Using IBM PC BASIC,"** San Francisco, Dellen Publishing Co., 1986.

- Shelly, Gary B., and Cashman, Thomas J., **"BASIC for the IBM PC,"** Boston, Boyd & Frazer Publishing Company, 1988.

- Simpson, Alan, **"Data File Programming on Your IBM PC,"** Berkeley, SYBEX, 1984.

- Sondak, Norman, and Hatch, Richard, **"Using BASIC on the CYBER,"** Chicago, Science Research Associates, 1982.

- Sondak, Norman, and Hatch, Richard A., **"Using BASIC on the IBM Personal Computer,"** Chicago, Science Research Associates, 1985.

- Spear, Bob, **"BASIC: Programming Fundamentals and Applications,"** Columbus, Merrill Publishing Company, 1987.

- Sternberg, Charles D., **"IBM Programs for Business,"** Rochelle NJ, Hayden Book Co., 1983.

— Waite, Mitchell, and Pardee, Michael, **"BASIC Primer,"** Indianapolis, H. W. Sams, 1978.

— Wells, Timothy D., **"A Structured Approach to Building Programs : BASIC,"** Englewood Cliffs, Prentice-Hall, 1985.

— Worlaind, Peter B., **"Introduction to BASIC Programming : A Structured Approach,"** Boston, Houghton Mifflin, 1979.

— Zage, Wayne M., **"Programming with MICROSOFT BASIC,"** Englewood Cliffs, McGrew-Hill Book Company, 1985.



١٤١٠هـ

● المؤلفان في سطور ●

■■ الدكتور غازي إسحق الخطيب

ـــ من مواليد عمان ـــ الأردن.

ـــ حصل على درجة الدكتوراه فى الحاسب الآلى من جامعة ميسيسبى، ولاية ميسيسبى عام ١٩٨٤م.

ـــ يعمل حاليا أستاذا مشاركا ـــ جامعة هاوارد ـــ واشنطن.

ـــ من خبراته العملية :

- أستاذ مساعد ـــ معهد الإدارة العامة.
- أستاذ مساعد ـــ جامعة ميسيسبى ـــ ولاية ميسيسبى ـــ أمريكا.

ـــ من أهم أعماله العلمية المنشورة :

- مقارنة بين قواعد البيانات من ناحية الأمن والسرية ـــ المؤتمر والمعرض الوطنى التاسع للحاسب الآلى ـــ الرياض ـــ ١٤٠٧هـ. بالاشتراك مع عزمى العيسى.
- دور الحاسب الآلى فى التنمية الإدارية ـــ العدد (٥٠) ـــ مجلة الإدارة العامة ـــ معهد الإدارة العامة ـــ الرياض بالاشتراك مع د. إبراهيم عبدالسلام، وعبدالعزيز القويز.
- تقييم أداء الحاسبات الآلية فى المراكز الحكومية فى المملكة العربية السعودية ـــ ندوة ـــ معهد الإدارة العامة بالاشتراك مع د. إبراهيم عبدالسلام، وكامل المبارك.

■■ الأستاذ عزمي حسن العيسى

ـــ من مواليد ارتاح ـــ فلسطين.

ـــ حصل على درجة الماجستير فى الحاسب الآلى (تحليل وتصميم النظم) من جامعة بتسبرج ـــ ولاية بنسلفانيا عام ١٩٨٤م.

ـــ يعمل حاليا محاضرا بمعهد الإدارة العامة.

ـــ من خبراته العملية :

- محلل نظم، شركة افاثار الاستشارية لنظم المعلومات بتسبرج.
- محاضر، كلية بوينت بارك ـــ بتسبرج.
- مبرمج، المجلس الوطنى للتخطيط ـــ عمان ـــ الأردن.
- مشغل، البنك التجارى الكويتى ـــ الكويت.

ـــ من أهم أعماله العلمية المنشورة :

- مقارنة بين قواعد البيانات من ناحية الأمن والسرية ـــ المؤتمر والمعرض الوطنى التاسع للحاسب الآلى ـــ الرياض ١٤٠٧هـ بالاشتراك مع د. غازى الخطيب.
- نحو تصميم شبكة اتصالات بأقل التكاليف فى المملكة العربية السعودية ـــ المؤتمر والمعرض الوطنى التاسع للحاسب الآلى ـــ الرياض ١٤٠٧هـ. بالاشتراك مع د. نزيه الدرينى وعبدالله باتو.

